International Society of Parametric Analysts

Parametric Estimating Handbook©

Fourth Edition – April 2008

International Society of Parametric Analysts

Parametric Estimating Handbook, Fourth Edition

ISBN # 0-9720204-7-0

.

ISPA/SCEA Joint Office
527 Maple Avenue East, Suite 301
Vienna, VA 22180
703-938-5090
Fax 703-938-5091
www.ispa-cost.org

Table of Contents

CHAPTER 6 COMPLEX SOFTWARE MODELS

CHAPTER 7 GOVERNMENT COMPLIANCE

List of Figures

Preface to the Fourth Edition

This Parametric Estimating Handbook is a comprehensive, "all in one place" guide for Industry and Government acquisition professionals who prepare, evaluate, or negotiate proposals which use parametric estimating techniques or for project stakeholders who wish to gain a better understanding of the application and use of parametric cost estimating. This handbook also serves as a foundation for companies that want to make more use of parametric tools in developing the basis of estimate (BOE) for their proposals to the Government or otherwise expand their use of parametrics.

Fourth Edition Updates

This Fourth Edition of the handbook updates the Third Edition (Spring of 2003). The Third Edition was not a comprehensive update. The motivation for the Third Edition was a general edit, a refresh for certain technical data, and an update to the various web sites that reference parametric applications. The Second Edition (Fall of 1999) replaced the First (Fall of 1995) by adding technical information and new chapters concerning Government regulations, implementation teams, and technical evaluations. The Second Edition also incorporated results from the Parametric Estimating Reinvention Laboratory.

This Fourth Edition is an extension of the third. The Third Edition was primarily a cosmetic and slightly modernized version of the Second Edition. The Fourth Edition includes new material and rewrites with the best parts of the Third Edition retained.

The focus of this edition is on process and benefits with less emphasis placed on tools and detailed mathematics. Although tools and math are important, the reader/practitioner is encouraged to go to the sources for detailed information about understanding, training, use, and other special topics (e.g., calibration/validation). Although such topics are discussed in the Fourth Edition, they are discussed primarily in the context of process. The goal was to make this edition more readable and understandable. All in all, we believe this edition to be superior in both content and quality to all previous editions.

About the International Society of Parametric Analysts (ISPA)

ISPA (www.ispa-cost.org) was founded more than 25 years ago. The first national conference was held in 1979. The genesis of ISPA began in 1978 at the third PRICE Users meeting in San Francisco in April of that year. At that meeting, ISPA was founded as an "international" organization. The first ISPA "Town Meeting" was in April, 1979. Three hundred participants from international locations attended and it was at this meeting that the ISPA was organized.

The *PRICE User's Bulletin (PUB)* was replaced by the *ISPA News* (premiered in 1981) with Charley Hopkins as Editor. Other ISPA publications include:

- Parametric World;
- Membership Directory;
- Journal of Parametrics;
- Conference Proceedings;
- This Parametric Estimating Handbook;
- Training materials based on the handbook.

ISPA has always provided excellent conferences and educational programs for its members. Truly international in scope, the ISPA conferences are held annually, and each leap year are at an international location.

In the estimating process, parametricians understand the technical and other cost drivers, thus making the parametrician a valuable member of a proposal team. The goal of ISPA is to continue to support parametricians throughout their career by stimulating tool development and by encouraging professional contributions. ISPA will continue to be a powerful force within the estimating community in the foreseeable future.

Acknowledgements

The International Society of Parametric Analysts (ISPA), as well as the general Industry/Government Estimating Community would like to thank the following people and organizations for their support during the effort to compile this edition of the Parametric Estimating Handbook. We apologize if anyone was overlooked.

- The ISPA Board of Directors for their overall support and funding.
- The Content Writing and Review Team consisted of the following people:
    - David Eck, DCAA;
    - John C. Smith, DCAA;
    - Jason Dechoretz, MCR;
    - Rich Hartley, USAF;

  - Jerry McAfee, DCAA;
  - Dan Ferens, USAF (retired);
  - Evin Stump, Galorath;
  - Dan Galorath, Galorath;
  - Bob Hunt, Galorath;
  - Neil Albert, MCR;
  - Hank Apgar, MCR;
  - Sherry Stukes, NASA JPL;
  - Dick Basner, Northrop Grumman;
  - Cheryl Saul, DCMA;
  - Keith Flaherty, DCMA;
  - Michelle Auwaerter, DCAA;
  - Dennis Lowery, USA, AMCOM;
  - Bruce Fad, PRICE Systems;
  - George Teologlou, PRICE Systems;
  - Madeline Ellis, Boeing;
  - Tony Fettig;
  - Bill Brundick, WEB Enterprises;
  - George Barbic, Lockheed Martin;
  - Steve Book, MCR.
- Joan Ugljesa (MCR) and Diane Welschmeyer (DCAA) for general editing and publishing support.

Parametric Estimating Handbook Sources

Electronic copies of the current edition of the Parametric Estimating Handbook can be obtained from the following sources:

- ISPA Web Site at
  www.ispa-cost.org
- SCEA Web Site at
  www.sceaonline.net
- DoD Acquisition Deskbook Web Site at:
  www.deskbook.dau.mil
- NASA Web Site at:
  www.jsc.nasa.gov/bu2/PCEHHTML/pceh.htm

- Galorath Incorporated
  www.galorath.com

User Comments

As in previous editions, this Fourth Edition of the Parametric Estimating Handbook is intended to be a living document. We encourage all handbook users to provide suggestions for improvement using the comment form at the end of the handbook. You may also go to the ISPA web site (www.ispa-cost.org) to make comments.

Introduction

About This Handbook

The detailed guidance, case studies, and best practices contained in this handbook are designed to provide an understanding of the "how-to" of parametric estimating. It is designed to help those involved in the acquisition process to become more familiar with parametric estimating as well as the techniques and tools used in the process. It is also designed to assist practitioners and managers involved in the processes to better understand and apply the tools and techniques to "real world" cost estimating problems.

People new to the parametric estimating practice will find this document to be an invaluable aid in the execution of their assignments. This handbook provides information about parametric estimating techniques, guidance on the acceptable use of tools, and methods for process and parametric estimate development and evaluation. The chapters mirror the process an organization may use in developing a parametric estimating capability.

This handbook presents and summarizes the best practices and lessons learned which an organization needs to know to successfully establish and utilize parametric estimating tools and techniques. This handbook also helps companies address the feasibility of using parametric techniques before they are implemented. Some of the critical feasibility issues assessed include:

- Availability of relevant historical data;
- Reliability of other estimating techniques versus parametrics;
- Costs versus benefits;
- Industry and Government support.

The Federal Acquisition Regulation (FAR) states that parametric estimating is an acceptable method for preparing proposals based on cost or pricing data, or data other than cost or pricing data. The primary benefit of developing a parametric estimating capability is that it streamlines the estimating and proposal process for both Industry and Government. Integrated product teams (IPTs), for example, have demonstrated that properly calibrated and validated parametric estimating techniques improve customer satisfaction (see Appendix J).

The objectives of this handbook are to help users to:

- Enhance and improve the quality of their estimates;

- Add more tools and techniques to their estimating toolbox;
- Provide internal and external estimate reviewers additional credibility and realism to their estimates;
- Reduce contract award cycle time;
- Reduce proposal preparation, evaluation, and negotiation costs.

About Parametric Estimating

Parametric estimating is a technique that develops cost estimates based upon the examination and validation of the relationships which exist between a project's technical, programmatic, and cost characteristics as well as the resources consumed during its development, manufacture, maintenance, and/or modification.

Parametric models can be classified as simple or complex. For this handbook, simple models are cost estimating relationships (CERs) consisting of one cost driver. Complex models, on the other hand, are models consisting of multiple CERs, or algorithms, to derive cost estimates.

Ancient History

Cost estimating has a very ancient history. It is even Biblical. Luke 14: 28 - 29 discusses the importance of "...[He should] sitteth down first, and counteth the cost, [to see] whether he have sufficient to finish it."

The question is, then, not whether an estimate should be prepared, but which approach should be used to estimate the cost of a specific application. The answer is, "it depends." This handbook will demonstrate that there is a place for parametric tools in the estimator's tool box. It will also answer the question about when and how parametric tools should be applied. The answers may be surprising to some people, because all too often parametric tools are not considered when perhaps they should be.

In Keith Burbridge's book, *A Touch of History,* Burbridge discusses the use of parametric estimating through the ages from Hero to present times. Indeed, it makes total sense that the efficiencies of parametrics would be recognized in some of the earliest recorded histories. Even today, who hasn't had a house appraised through what is a true parametric application? Is every piece of drywall, number of bricks, and each two by four counted and estimated? The answer is, "no." Another technique is used. The parametric application considers such things as house size (square feet of living space), style of house, condition, location, and even the zip code. A "formula" with such input variables then predicts a house value. This is a universal parametric application. And there are many other applications this handbook describes.

Current Industry and Government Practices

Industry and Government practitioners commonly use parametrics to perform a variety of analyses, such as independent cost estimates and trade studies. Practitioners (users) have argued that proposal preparation, evaluation, and negotiation costs and cycle time can be reduced considerably through the increased use of parametric estimating. They also stated that these benefits can be achieved while maintaining or improving the quality of the estimates produced.

Industry saw the need to team with the Government to demonstrate that parametrics are an acceptable and reliable estimating technique after achieving some success with the broader uses of parametric techniques (e.g., independent estimates and trade studies). In December 1995, the Commander of the Defense Contract Management Agency (DCMA) and the Director of the Defense Contract Audit Agency (DCAA) sponsored the Parametric Estimating Reinvention Laboratory under the auspices of the Parametric Cost Estimating Initiative. The purpose of the Parametric Estimating Reinvention Laboratory was to test the use of parametric estimating techniques on proposals and recommend processes to enable others to implement these techniques. The primary objectives of the Parametric Estimating Reinvention Laboratory included:

- Identifying opportunities for using parametric techniques;
- Testing parametric techniques on actual proposals submitted to the Government;
- Developing case studies based on the best practices and lessons learned;
- Establishing formal guidance to be used by future teams involved in implementing, evaluating, and/or negotiating parametrically based estimating systems or proposals (e.g., this handbook).

Thirteen Parametric Estimating Reinvention Laboratory teams tested and/or implemented the full spectrum of parametric techniques. The Industry and Government teams used these techniques to develop estimates for a variety of proposals, including those for new development, engineering change orders, and follow-on production efforts.

The estimates covered the range of use from specific elements of cost to major-assembly costs. The teams generally found that using parametric techniques facilitated rapid development of more reliable estimates while establishing a sound basis for estimating and negotiation. In addition, the teams reported proposal preparation, evaluation, and negotiation cost savings of up to 80 percent, and reduced cycle time of up to 80 percent.

The contractors, with feedback from their Government team members, updated or revised their estimating system policies and procedures to ensure consistent production of valid data and maintenance of the tools employed. The Parametric Estimating Initiative Closure Report (see Appendix G) provides details on the best practices for implementing parametric techniques. The lab results have also been integrated throughout this handbook in the form of examples, best practices, and lessons learned with respect to implementing, evaluating, and negotiating proposals based on parametric techniques.

Beginning in the early and mid 1990's, both Industry and Government, through the Parametric Cost Estimating Initiative (PCEI), evaluated the ability of parametric estimating techniques and tools to support Government proposal cost estimating requirements. The key issues considered included parametrics' ability to maximize the use of historical data in the estimating process, increase estimate realism, and reduce the costs associated with proposal preparation, evaluation, and negotiation. The PCEI, through numerous workshops and the Parametric Estimating Reinvention Laboratory, concluded that parametric techniques and tools, when properly implemented and correctly used, could produce realistic cost estimates at significantly reduced costs and times.

As a result, contractors today generally use parametric techniques to improve their Government contracting practices, as well as the quality of their estimating. Although Industry and Government use of parametrics can be somewhat different (e.g., the Government may use parametric tools for independent estimates more than Industry), there will still be significant overlap in application. The use of integrated product teams (IPTs) enhances the implementation of parametric tools (see Appendix J).

Genesis of This Handbook

Contractors use a variety of techniques to develop estimates, the most frequently employed being analogous, bottoms-up, and parametric estimating.

A primary responsibility of a project cost estimator is to select the estimating methodology that most realistically estimates program costs, while making the most economical use of the organization's estimating resources. With respect to this requirement, the PCEI identified two general concerns about the use of parametric tools and techniques, and their ability to adequately support cost estimating requirements for contracting proposals.

First, the Truth in Negotiations Act (TINA) data issues seems to be the greatest concern regarding the use of parametric estimating methods. TINA requires that cost or pricing data be certified as current, accurate, and complete as of the date of negotiation or another agreed-to date as close as practicable to the date of negotiation. TINA also requires contractors to provide (disclose) to the Government all the facts available at the time of certification, or an otherwise agreed-to date. Parametric tools should demonstrate that, when properly calibrated and validated, comply with the requirements of TINA.

Second, the use of statistical representations of historical data as a basis of forward estimates was a major concern for an estimating culture that developed and reviewed reams of paperwork in a bottoms-up environment. This cultural issue was much harder to resolve and required the publishing of this handbook and the development of professional training programs. Thus, publishing this handbook was a top priority of the PCEI.

Benefits of Using Parametrics

The benefits of using parametrics are well documented. It is estimated that the savings to proposal preparation is between 40 percent and 80 percent as compared to the "normal" bottoms-up approach. Parametric tools and techniques have much more versatility than other estimating approaches. There are numerous reasons for this. Here are a few:

- Better estimates are provided, often in a matter of minutes;
- There exists a high-quality link between the technical and cost proposals;
- The data is well understood through the calibration and validation activities;
- It is much easier to estimate conceptual designs;
- Early costing cannot be done effectively any other way;
- No bill of material (BOM) is required;
- It is much easier to handle scope, technical, and performance changes.

Parametrics in Support of CMMI Certification

One of the emerging benefits of parametric estimating is in the Software Engineering Institute's (SEI's) Capability Maturity Model Integration (CMMI®) certification. CMMI is a process by which contractor organizations are evaluated against a standard set of process and business measures established by an Industry steering committee working through the auspices of Carnegie Mellon University. The intent is for contractor organizations to certify themselves against a selected CMMI model. There are many models to choose from depending on the type of organization and the type of products that they develop.

Obtaining a CMMI maturity level (1 to 5, with 5 being the highest) through the audit process provides the organization a measure of how mature and effective their processes and business practices are against the CMMI standards. The certifications are sought-after as discriminators in competing for new business opportunities. Chapter 6 discusses the application of CMMI principles to the software estimating environment.

The CMMI standards apply to the estimating process as well and highlight areas of the process where specific characteristics must be present to achieve certification. The higher the certification, the more rigorous the estimating process must be. Some of these characteristics are interpreted differently at different levels of certification, and by different auditors, but in general CMMI addresses the following estimating characteristics:

1. An estimating process must identify and employ a documented method for estimating software, hardware, and so forth including the use of work products and task attributes.

2. An estimating process must be supported with the necessary historical data and organizational databases and programs must use the organizational historical data for estimating.
3. An estimating process must include monitoring actual values of the work product and task attributes against the estimates to assess the quality of the attributes used in the estimating process.

The CMMI and the parametric estimating process should have the following characteristics:

- Estimates should rely on historical data;
- Judgments and modifiers are acceptable elements, but historical context is required;
- Historical data are accessible through databases/repositories;
- Process is consistent and repeatable;
- Process is monitored for improvements/learning;
- Used consistently across the enterprise.

Handbook Outline

The general content of this edition of the handbook is as follows:

Chapter 1 Parametric Analysis Overview

This chapter describes the parametric estimating process as well as the procedures and techniques for implementing that process as seen from a management perspective. As well as process, the chapter describes the various types of parametric applications, the organization required to implement the techniques, use of cost IPTs, the necessary competencies, and roles and responsibilities in the parametric estimating organization.

Chapter 2 Data Collection and Analysis

Chapter 2 discusses the methods and techniques of data collection and analysis for use in the development of CERs and more complex parametric models. The chapter also discusses the techniques of data capture and normalization. Detailed technical math can be found in Appendix B. There is an emphasis on real world examples.

Chapter 3 Cost Estimating Relationships

Chapter 3 discusses the development and application of cost estimating relationships (CERs). As with Chapter 2, detailed technical math can be found in Appendix B.

Chapter 4 Complex Company Developed Complex Models

Chapter 4 describes the development and application of company-specific complex models. These are also known as "in-house" developed models. This chapter expands the topic of CERs into multi-variate and multiple use CER models.

Chapter 5 Complex Hardware Models

This chapter expands the topic of CERs into the realm of complex hardware tools and models (those that are commercially available for license). It covers the topics of hardware model development, application types, calibration/validation, strengths, weaknesses, and model content.

Chapter 6 Complex Software Models

This chapter continues the expansion of the topic of CERs into the realm of complex software tools and models (those that are commercially available for license). It covers the topics of software model development, application types, calibration/validation, strengths, weaknesses, and model content.

Chapter 7 Government Compliance

This chapter defines the U.S. Government oversight requirements necessary to ensure parametric estimating systems' compliance with all laws and regulations. It includes TINA, FAR, DFARS, and CAS citations as well as DCMA technical evaluation criteria, and DCAA audit criteria.

Chapter 8 Other Uses of Parametric Tools

Chapter 8 is devoted to the various other parametric applications that make these tools and techniques invaluable to the practitioner and program manager. These parametric applications are in addition to the basic cost estimating function. Examples are included.

Chapter 9 International Use of Parametrics

This chapter describes how the international community uses parametric estimating tools and techniques, and how that use differs from that in the United States.

Appendices and Other Attachments

The appendices contain the informative and useful adjunct materials not fully defined or described in the main body of the handbook. The appendices include:

- Appendix A which provides a description of modelers and model builder tools that are available for license;

- Appendix B which provides the detailed technical math of the cost estimating relationships;
- Appendix C, frequently asked questions;
- Appendix D, related web sites and supplementary information for other parametric resources;
- Appendix E, parametric estimating checklists;
- Appendix F, useful information to include in a memorandum of understanding;
- Appendix G, the Parametric Cost Estimating Initiative Closure Report;
- Appendix H, the Space Systems Cost Analysis Group risk summary;
- Appendix I, the Space Systems Cost Analysis Group, Nonrecurring and Recurring Cost Definitions and Cost Accounting Guidelines;
- Appendix J, a discussion on establishing a parametric implementation team;
- Appendix K, a discussion on preparing subsystem level datasheets.

Other attachments include:

- Glossary (list of acronyms and definition of terms);
- References;
- Handbook User Comments form.

Part One
The Basics of Parametric Analysis

CHAPTER 1

Parametric Analysis Overview

The purpose of this chapter is to describe, for those who are not intimately involved in the parametric costing process, the basic knowledge required to manage and support that process. Many support personnel do not need an in-depth knowledge. These people include project and other managers, proposal reviewers, pricers, accountants, and other knowledgeable professionals who come into contact with parametric analyses from time to time during their careers.

1.1 Best Practices of the Parametric Analysis Process

Parametric analysis as practiced by members of the International Society of Parametric Analysts (ISPA) involves computerized cost models that use the parameters of consequential projects and the project's products to estimate the use of resources required to perform the project such as labor, materials, and time. These models have economic value because, properly designed and used, they can improve the accuracy of project estimates, reduce the likelihood of serious overruns of budgets and schedules, reduce the cost of preparing project proposals, and enable project leaders and stakeholders to consider more options with regard to the best way to proceed.

Many parametric models also serve to advise on the uncertainties and risks associated with project costs and schedules. This is an important function, because modern projects are often enormously complex. Uncertainties and risks may effect profound changes. A purely cost or duration estimating model will provide what is called a "point estimate" of cost or duration. A point estimate is a single number that will always be in error to a greater or lesser extent.

A model that deals with uncertainty and risk will provide a "range estimate," also called a probability distribution, that is, an estimate that tries to give some idea of the possible range of cost or schedule outcomes, and of the relative likelihood of particular outcomes. The process of developing range estimates are inherently of more value to project management and stakeholders than point estimates because they help with understanding of what could happen and why. They frequently point to certain risk abatement possibilities and options that otherwise would have gone unnoticed.

Construction and use of valid parametric models is not free. For that enterprise to be worthy, its effectiveness must at least equal its costs. Over the years, parametricians have generally succeeded in demonstrating the cost effectiveness of their work. The profession has grown steadily in both number and quality of practitioners. That has happened through a gradual process of convincing project stakeholders of the integrity of the modeling process, and of the increasing professionalism of model users.

The objective of this chapter is to provide guidance for managing the parametric analysis process. It describes best practices associated with what we will formally call the parametric analysis process. This process is not monolithic in the sense of massive, totally uniform in character, and slow to change. It is an evolving process, benefiting from continuous improvements and new ideas. Experience has shown that certain aspects of the process are best done in particular ways if the best results are to be obtained for all concerned. Those best ways is the focus of this chapter.

The process has three major components: database development, model development, and model use. In most situations, the same people do all parts of the process. This is especially the case when a company, or a Government agency, decides that they should build a parametric model (or models), for their own, often specialized, internal use. In other cases, organizations may decide to license or otherwise acquire a commercially available general or special purpose model that they believe will adequately serve their needs.

Parametric analysis is a major management innovation. In common with several other management innovations, such as network scheduling, earned value analysis, and many of the methods of operations research, modern parametric analysis had its genesis in the U.S. and British military-industrial complex. ISPA's present membership is still associated with and heavily influenced by that world, but its sphere of interest now includes other U.S. Government agencies, and additional companies and Governments in Europe, Australia, and in Asia.

Use of parametrics also has spread to the commercial world, especially to the construction industry and to companies that build or buy software, and that is now a growing share of the business for companies that produce commercial models still used primarily by ISPA members. Nevertheless, the best practices discussed here are heavily influenced by the needs of Government, U.S. and other. That should be kept firmly in mind. Some of the practices might not apply, or might be less rigorous, if it were not for the need to maintain great openness and integrity in the handling of public money.

The Government interest in parametric best practices strongly affects the construction and use of in-house parametric models. It also affects the use of commercially available models and to a lesser extent their construction. In-house developed cost models used in Government procurements generally must be open to and approved by the Government, at least in some respects. Commercially built models, on the other hand, are generally proprietary, at least in part, and their customers use them because they offer some economic advantages. Nevertheless, their use in estimating Government-paid costs is carefully scrutinized by the Government, and users must generally show that the models have been calibrated to a particular project environment and way of doing business.

This chapter will first address database development, then the model building part of the process, and then using the model. In each instance, the appropriate steps will be described in a simplified flowchart format. Then, each step of the flowchart will be discussed. The level of detail of the discussion often will be limited because subsequent chapters and appendices in this handbook provide much more detail.

1.1.1 The Parametric Model Building Process

Figure 1.1 illustrates the typical steps of cost model development as performed by the parametrician.

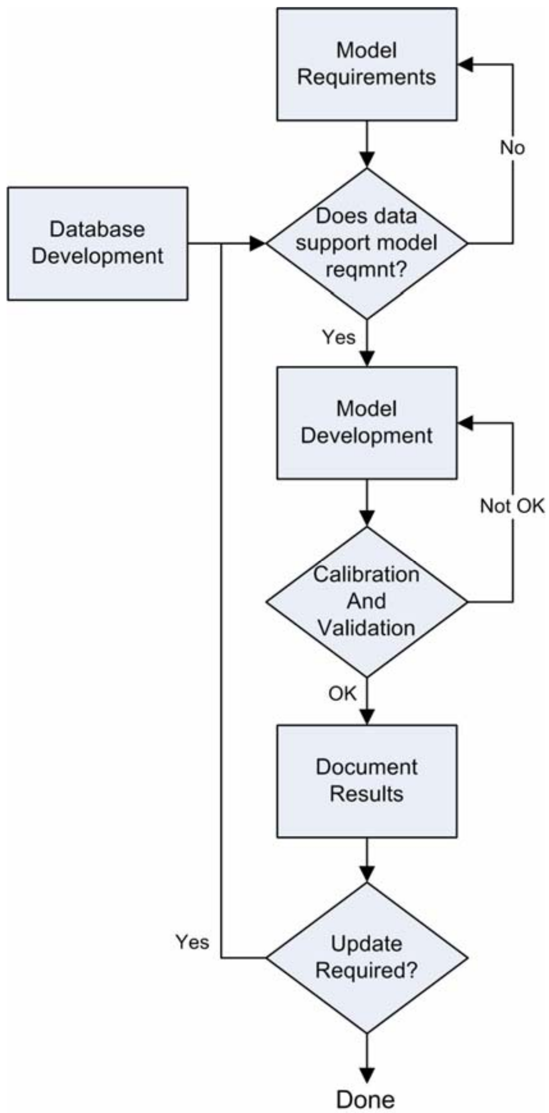


Figure 1.1 Flowchart of Cost Model Development by the Practitioner

It should be noted that the flow of work for commercial model development and the development of in-house models differ. In-house models that are developed for a specific purpose are strongly tied to the available historical data. Thus, the development and normalization of auditable historical data is the key starting point. The historical data will dictate and restrict the level of detail and the approach the parametrician can take.

In-house models discussed here tend to be for a specific purpose, such as a new business competition, and not to support a family of acquisitions. Although, the in-house model could be used for other acquisitions with tailoring to a work breakdown structure (WBS), phasing, quantities, hardware specifics, and so forth.

The development of unique in-house cost models as experienced by the parametrician occur through these steps:

- Database development;
- Model Requirements;
- Resolution of model architecture and data availability;
- Model development;
- Model calibration and validation;
- Model documentation;
- Model updating.

1.1.1.1 Database Development

A sound database is key to the success of the parametrician. A cost model is a forecast of future costs based on historical fact. Thus, future cost estimates must be consistent with historical data collection and cannot provide a lower level of detail than provided by the historical detail without some allocation or distribution scheme devised by the parametrician.

Parametric techniques require the collection of historical cost data (including labor hours) and the associated non-cost information and factors that describe and strongly influence those costs. Data should be collected and maintained in a manner that provides a complete audit trail with expenditure dates so that costs can be adjusted for inflation. Non-recurring and recurring costs should be separately identified. While there are many formats for collecting data, one commonly used by industry is the WBS, which provides for the uniform definition and collection of cost and certain technical information. If this is not the case, the data collection practices should contain procedures for mapping the cost data to the cost elements of the parametric estimating technique(s) which will be used.

The collection point for cost data is generally the company’s financial accounting system, which in most instances contains the general ledger and other accounting data. All cost data used in parametric techniques must be consistent with, and traceable to, the collection point. The data should also be consistent with the company’s accounting procedures and generally accepted cost accounting practices.

Technical non-cost data describe the physical, performance, and engineering characteristics of a system, sub-system, or individual item. For example, weight is a common non-cost variable used in cost estimating relationships (CERs) and parametric estimating models. Other examples of cost driver variables are horsepower, watts, and single lines of software code. A fundamental requirement for the inclusion of a technical non-cost variable in a CER is that it must be a significant predictor of cost.

Technical non-cost data comes from a variety of sources including the MIS (e.g., materials requirements planning (MRP) or enterprise resource planning (ERP) systems), engineering drawings, engineering specifications, certification documents, interviews with technical personnel, and through direct experience (e.g., weighing an item). Schedule, quantity, equivalent units, and similar information come from industrial engineering, operations departments, program files, or other program intelligence.

Other generally available programmatic information that should be collected relates to the tools and skills of the project team, the working environment, ease of communications, and compression of schedule. Project-to-project variability in these areas can have a significant effect on cost. For instance, working in a secure facility under “need to know” conditions or achieving high levels in various team certification processes can have a major impact on costs.

Once collected, cost data must be adjusted to account for the effect of certain non-cost factors, such as production rate, improvement curve, and inflation; this is data normalization. Relevant program data including development and production schedules, quantities produced, production rates, equivalent units, breaks in production, significant design changes, and anomalies such as strikes, explosions, and other natural disasters are also necessary to fully explain any significant fluctuations in the data. Such historical information can generally be obtained through interviews with knowledgeable program personnel or through examination of program records.

As new business opportunities or parametric model applications materialize it may be necessary to add to the database through a formal data collection of relevant program history, or it may be necessary to reduce the database to a subset of more appropriate historical data points, eliminating irrelevant historical programs.

1.1.1.2 Model Requirements

The expectation of a parametric model is that it will estimate costs virtually instantaneously and accurately if the correct information is entered with respect to its parameters. It can do this repeatedly without deviation. Generally, there is an even higher expectation, namely that a parametric model will do these things quicker and better than alternative methods, such as bottoms-up estimating or detailed analogy estimating. This is especially true if the model is intended to support numerous cost trade studies and analyses. If that is not true, the expense of building a parametric model may not be justified.

While crude analogy estimates can sometimes be produced in minutes, they are not famous for their accuracy. More detailed analogy estimates can be quite accurate, but they are usually time consuming to build. Bottoms-up estimates are notoriously inaccurate very early in project planning because of poor understanding of project scope, but typically improve as time goes on, and a bill of materials (BOM) is built. They are usually very time consuming and expensive. A well conceived and constructed parametric model offers rapid, inexpensive estimating at any stage of project life, and is generally the more accurate method in the early days of a project.

The scope of the model is strongly dictated by the database and the specification for the model. The specification is generally a function of a request for information (RFI), request for proposal (RFP), or other official Government request, or this model may even be by request of management in anticipation of a new business opportunity. In any event, the level of detail required by the model will be a function of the information desired tempered by the nature of the data available in the database, the time-frame required for developing the model, and so forth.

The in-house model is typically designed to estimate a system such as a communication satellite system, land-based missiles or armored tanks, a particular type of hardware or software such as a battery or fire control system, or perhaps a particular function, such as systems engineering, and may be limited to development costs only, or production costs only. Many in-house models give "most likely" point estimates, but there is a significant trend within industry to provide range estimates based on risk and cost uncertainty. The new parametric model may be best served by a combination of several commercial models tied together by in-house developed CERs and algorithms.

1.1.1.3 Resolution of Model Architecture versus Data Availability

What exactly is meant when we speak of the architecture of a parametric model? Keep in mind that today a parametric model is a software product. Fortunately, the architecture of most parametric models fits a fairly consistent process. There is input, there is output, and in between there is a collection of cost estimating relationships and perhaps other types of mathematics and logic.

Every parametric model contains at least one cost estimating relationship, more commonly known in the parametrics community as a CER. A CER is always a mathematical relation, and always involves numbers, but otherwise can take several forms.

The most common forms are:

- Algebraic equations;
- Lookup tables.

Other forms are possible in certain types of very specialized models, but we will not attempt to list them here. The most general expression for the algebraic equation form of a CER is $y = f(x_i)$. Here, y represents a desired estimate, usually in currency units (e.g., USD), labor hours expended, or periods of time consumed (e.g., months). Mathematicians would refer to y as a dependent variable. The f represents some functional relationship that could be almost anything, including linear equations and a variety of non-linear equations, such as polynomials, power laws, exponentials, and so forth. The key point about the selection of f is that the resulting equation must be a good "fit" to the supporting data. Assuming that is true, we also assume that it will produce estimates that reasonably represent what will happen in future real world projects.[1]

The x_i represents the possibility of more than one independent variable. Most commonly, those independent variables are the parameters that the model builder has chosen as parameters driving the dependent variable. Model users initiate estimating by entering known or at least suspected values for these parameters.

Lookup tables are basically mathematical functions that are expressed in tabulated form. Use of lookup tables is sometimes more convenient for commercial model builders than use of algebraic functions. This is particularly true for "discrete" drivers such as material of manufacture, number of axes of rotation, number of external interfaces, and so forth.

A key aspect of model architecture is the choice of parameters. A prime requirement for use of a particular parameter is that it must be either a direct cause of the level or amount of the resource being estimated, or must strongly correlate with it. An example of a direct cause is the number of optical elements in a telescope. The more of them that are required, the higher the cost. Their optical quality is another direct cause, as is their combined surface area.

A prime requirement for the selected cost driving parameters considered as a set is that the amount of correlation between any two of them should be small. Correlation is a statistical term. The most used measure of it is the coefficient of variation (R^2). The coefficient measures the strength and direction of a linear relationship between the two variables. Most spreadsheets will do this calculation.

A commonly used parameter in estimating many types of hardware is its weight. In many cases, weight does correlate strongly with cost, but it is seldom a direct cause. In fact, attempts to reduce weight can increase cost significantly. So why use weight as opposed to using more direct cost driving parameters? Weight is used because data for it is almost always available. Commonly on projects where weight is important, reasonably accurate weights are available very early, even in the proposal phase. But when using weight, it is virtually always necessary to use

[1] We can now define what could be called the Fundamental Assumption of Parametric Estimating. A fair wording for it is the following: ***"If carefully selected and adjusted historical project outcomes are fitted with sufficient accuracy by a set of mathematical relationships, then that same set of mathematical relationships will estimate with similar accuracy the outcomes of sufficiently similar future projects."*** Note the profuse use of qualifiers such as "carefully," "adjusted," "sufficient," and "similar." It is because of the need for such qualifiers that parametricians must exercise due diligence in selection and treatment of data, and careful testing of modeling concepts. The need for such qualifiers also prompts customers presented with parametric estimates to demand justifications of model construction and use.

other parameters as well, to prevent anomalous results such as incorrect decreases in cost when engineers attempt weight reduction programs (see inset below).

The Problem with Weight

Here is a simple, easily understood demonstration of the problem with using weight (alone) as a cost driving parameter.

A engineer specifies that a piece of round aluminum stock is to be used as a drive shaft for a large vehicle. It is 100 inches long and four inches in diameter. Using elementary geometry, its volume is approximately 1,256 cubic inches. Aluminum has a density of approximately 0.1 pounds per cubic inch, so the shaft weighs about 126 pounds. Assuming that aluminum costs $2/pound, the cost of the shaft is about $252.

The engineers find that the vehicle is too heavy, and a weight reduction program is ordered. The engineer in charge of the drive shaft carefully reviews his design margins and finds that he can get by with a 3.5 inch diameter drive shaft. Unfortunately, 3.5 inches is not a stock size, so the engineer specifies that the four inch shaft must be turned in a lathe and reduced in diameter to 3.5 inches.

It takes a skilled machinist one hour to do this work. His work station charges $100/hour for work processed there. So, the cost of the shaft is increased from $252 to $352, a 40% increase. But the weight of the shaft has decreased from 126 pounds to 96 pounds, a 25% decrease. To keep this anomalous result from happening, a parametric model builder who uses weight as a cost driver must somehow account for the extra effort of the machinist. This will require use of at least one additional input parameter.

With regard to users, timing, convenience, flexibility and ease of data availability are important issues. A major factor in favor of building and using parametric models is their ease of use early on with minimal project information, when many key decisions tend to be made. As noted above, one reason for the popularity of weight as a parameter is its early and easy availability. A frequent and vital part of that judgmental process is viewing and interpreting scatter plots.

Figure 1.2 shows ten hypothetical data points of cost, y, versus a direct cost driving parameter, x. Inspection of these points strongly suggests that a straight line is a good fit to the data, and a careful analyst would probably proceed with that assumption. The straight line that has been fitted to this data is based on a frequently used and relatively simple statistical model called ordinary least squares (OLS).

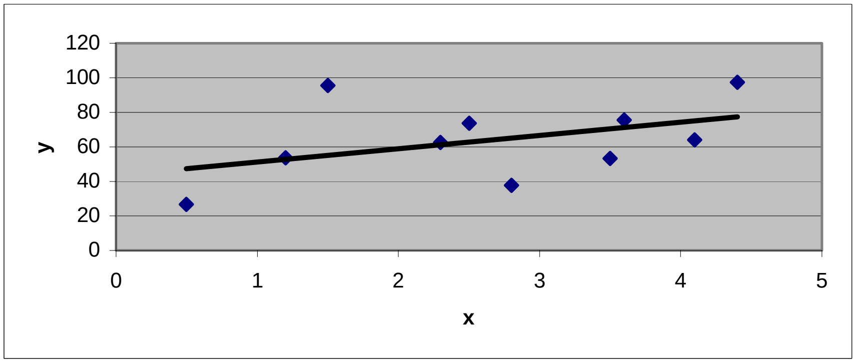


Figure 1.2 More Hypothetical Plot Points

The OLS provided best fit equation is y = 3.1772x + 1.8123. Obviously, the fit is not exact, because none of the points lie exactly on the line. Additional analyses, discussed in Chapter 2 of this handbook, are generally appropriate to analyze just how good the fit is. If it is good enough to satisfy the analyst and customers, a one independent variable CER can be declared. But what if the result is as shown in Figure 1.3? Here, the data scatter is much worse. What to do?

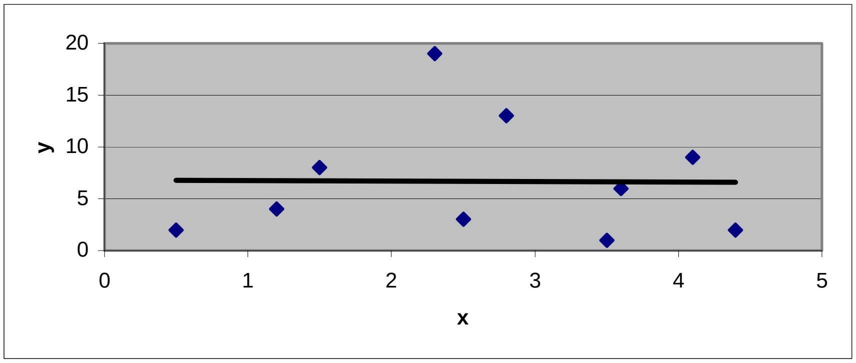


Figure 1.3 Poor Choice of Cost Driver

The first thing to do is to understand the several reasons why data scatter may be observed. The most prominent among them are:

- Poor choice of cost driving parameter;
- Presence of one or more other (but as yet unrecognized) cost driving parameters;
- Presence of non-normalized parameter values;
- Data collection errors;
- Inconsistent cost classification;
- Non-linearity of the x-y relationship.

We briefly discuss each of these in the order shown.

Poor Choice of Cost Driving Parameter

Engineers and others who work with a new product in a project environment generally have a good intuition for what parameters drive cost of the product. However, intuition can sometimes be deceived. The plot in Figure 1.3 is an extreme example. It should be noted that a parameter that is poorly associated with one project cost may be an excellent driver for a different project cost. For example, a driver that poorly predicts development labor hours could possibly be a very good predictor of production "touch" labor hours.

Presence of One or More Other Cost Driving Parameters

Most costs depend to some extent on more than one parameter. For that reason alone, scatter plots almost always contain scatter. One good way to detect this is to make scatter plots of other suspected drivers. Unfortunately, scatter plots with two or more independent variables are very difficult to make and to interpret, but usually dependence on more than one parameter can be detected with two dimensional plots.

Presence of Non-Normalized Parameter Values

Consider cost values for essentially the same kind and amount of material taken from two different projects, one done in 1999 and one done in 2003. In the 1999 project, the material cost $5/pound, but in the 2003 project it cost $5.50/pound. There could be more than one reason for this, but the most common reason is cost inflation. Inflation is one of a set of cost-affecting parameters commonly referred to as normalization factors. Certain cost data that comes from different years will typically be affected by inflation, and unless this effect is accounted for and corrected, the model can have significant errors.

It should be noted that different kinds of cost are not necessarily affected in the same way by particular normalization factors. For example, cost of a certain material such as aluminum may be strongly affected by inflation, while labor hours are not much affected. However, labor hours can be affected by industry productivity trends, "learning" phenomena, and other considerations such as skill mix.

Proper selection of normalization factors and the mathematics of normalization corrections are an area when analyst judgment and experience are important.

Data Collection Errors

Data collection errors can occur when the data are first recorded, and also when the parametric analyst acquires it from wherever it has been stored. If the original recording was in error, finding and correcting the error is often difficult.

Inconsistent Cost Classification

One of the most difficult tasks undertaken by parametric analysts is to sort out inconsistent data. The task is doubly difficult if the data comes from more than one organization, if accounting practices differ. Mergers and acquisitions within industry means data from disparate accounting systems have been combined. It would be foolish to assume that the contributing accounting systems were identical and that the data from the two was homogeneous. Experience, good judgment, and knowing the right questions to ask can be important to doing this well.

For example, an area of potential confusion is the distinction between recurring and non-recurring cost. In some companies, engineering development work is treated as a burden on production effort (recurring), while in others it is treated as a standalone non-recurring cost. See Appendix I for additional discussion on this topic.

What are the best practices with regard to possible data inconsistency?

- Recognize that it can happen, even within the same organization.
- Within your own organization, be sure you clearly understand the meanings of the various accounting classifications and how costs are assigned to them.
- Recognize that if data comes from more than one organization, some cost inconsistency is likely. Ask questions about their ways of treating costs.
- Try to bore down to the actual labor tasks that are done; work with labor hours to the extent possible, not labor costs (e.g., dollars).
- Recognize that even labor hours can have some inconsistency if the skill mix is changing.

Non-linearity of the X-Y Relationship

It would be simpler for parametricians if all costs were linearly related to their various drivers. Unfortunately, this is often a non-linear world and the assumption of linearity will not always work. For accuracy's sake, the non-linearity often must be accounted for.

Figure 1.4 shows a rather mildly non-linear x-y relationship. This plot was created using the popular MS Excel spreadsheet, which has the capability to do some simple "best fit" curve fitting to plotted data. While that capability should not be the basis of a formal CER design, it is an excellent tool for detecting non-linearity and making a preliminary assessment of the best non-linear function to use to fit the data.

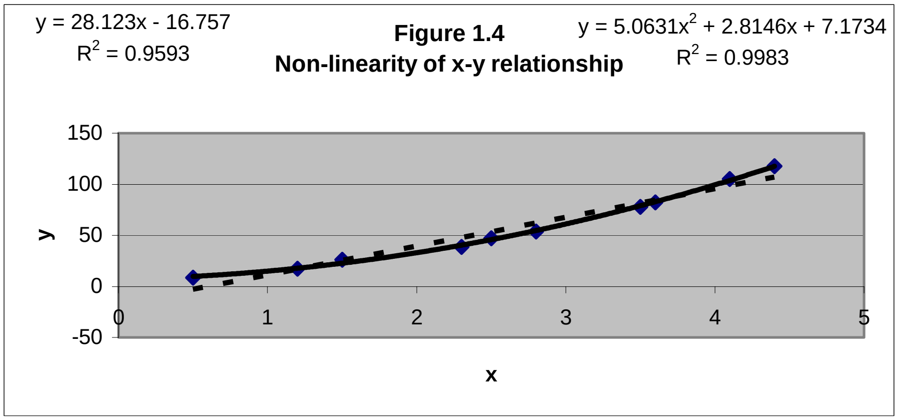


Figure 1.4 Non-linearity of X-Y Relationship

In Figure 1.4, the dashed line is a straight line, and it is not as good a fit as the curved line, which happens to be a second order polynomial, also known as a quadratic equation. The best fit equations of both curves appear at the top of the plot. Also appearing at the top of the plot is the statistic, R^2, the coefficient of variation. This statistic, which ranges from zero to one depending on how well the curve fits the data, is useful for rough comparisons of goodness of fit of different types of curves. Note that the better fit of the quadratic is confirmed by its higher R^2 value.

1.1.1.4 Model Development

During this phase, the team refines the scope of the model's requirements, and defines the methods and assumptions which establish the basis for its business rules and estimating relationships. User requirements and input/output interfaces are also identified.

The development of a complex model incorporates many anticipated uses and goals such as estimating/users' requirements, life-cycle costs, systems engineering costs, forward pricing rates and it must integrate these into the parametric estimating approach. The modeling process, in particular, focuses on these tasks:

- Specifying the estimating methods for accomplishing the estimating goals;
- Identifying the job functions and other elements of cost that will be estimated;
- Defining data input structures and WBS elements.

Complex models may contain a number of different estimating techniques (e.g., CERs, the input of discrete estimates), and must document how they all interact.

Development of a complex model is an iterative process. As more technical definition is understood and the baseline design detail increases, the elements of the parametric model expand to capture those new details. Thus the model continues to evolve from a simple CER-driven model to a complex CER-driven model.

1.1.1.5 Model Calibration and Validation

Parametric models should be calibrated and validated before they are used to develop estimates for proposals. Since complex models are based on an organization's historical data, they are considered to be self-calibrated. The validation process, however, applies to all parametric estimating techniques, whether CERs, complex models, or commercial models.

Validation is the process, or act, of demonstrating the complex model's ability to function as a credible estimating tool. Validation ensures:

- The model is a good predictor of costs;
- Estimating system policies and procedures are established and enforced;
- Key personnel have proper experience and are adequately trained.

The purpose of validation is the demonstration of a model's ability to reliably predict costs. This can be done in a number of ways. For example, if a company has sufficient historical data, data points can be withheld from the model building process and then used as test points to assess the model's estimating accuracy. Unfortunately, data sets available are often extremely small, and withholding a few points from the model's development may affect the precision of its parameters. This trade-off between accuracy and testability is an issue model developers always consider.

When sufficient historical data are not available for testing, accuracy assessments can be performed using other techniques. For example, a comparison can be performed between an estimate developed from a complex model and one prepared using other estimating techniques.

Another testing methodology compares a program's final cost to the complex model's estimate of it. However, it may be months, or years, before this approach can be applied to a given program. The model team may use this method when a program is near completion, or is at a point where a meaningful earned value performance index for it can be determined.

1.1.1.6 Model Documentation

Model documentation requires that configuration control of the assumptions, conditions, and changes to the model are recorded as they occur. Management will want to know the version of the model, including all changes and assumptions being used for costing.

1.1.1.7 Model Updates

Model updates are evolutionary. They are often updated on the fly as changes occur. All changes should be documented.

1.1.2 Parametric Model Use

The degree of rigor applied to the use of a parametric model should be no less than that applied to its construction. The steps of parametric model use are:

- Special settings;
- Model calibration;
- Parameter values and ranges;
- Using the results.

It should be noted that the flow of work need not always conform exactly to the sequence shown in the flowchart (Figure 1.1). Sometimes it may be possible, depending on resources available, to do in parallel, or partly in parallel, or even out of order, some of the tasks that are shown to be in series.

1.1.2.1 Special Settings

Many parametric models have special global settings that influence the overall response of the model. Common examples of such settings are year of inflation, inflation table to be used, method and slope of learning, average labor rates, average overhead rates, project start dates, and so forth. If a model has global settings, they must all be checked and properly set before estimating activities begin.

1.1.2.2 Model Calibration

Most simple CER models do not have a calibration capability nor do they need calibration if they are based on company history. The more sophisticated complex models have a calibration capability. The main reason they do is that the data they are based on may come from various sources and represents "industry typical" conditions. These conditions may be somewhat different than the conditions in a particular organization (more on this in Section 1.3). Users therefore need to adjust their model to accommodate the differences. These adjustments typically are not huge. Huge differences are unlikely in competitive markets, but they can occur in non-competitive situations, or in situations where an organization has a decided advantage, as for example in the availability of high quality but inexpensive labor (e.g., in university graduate schools). The topic of calibration is discussed in additional detail throughout this handbook.

1.1.2.3 Parameter Values and Ranges

To produce a cost result, the user must enter values for all parameters required by the model. In some models, particularly the simpler CERs, the user must enter values for every parameter. In others, particularly the more sophisticated complex models, the model may enter some default parameter values for the user, based on user selection from among a few basic estimating categories. The presence of such default "presets" in complex models is a labor saving device for users. The user can either accept or reject the preset defaults.

The simpler CERs typically require entry of only a single value for each parameter. Some of the more sophisticated complex models require several entries for each parameter for purposes of performing a risk analysis. One of these entries is typically what the user perceives to be the most likely value. The other entries are uncertainty parameters based on the probability distribution assumed.

Models that require uncertainty inputs are equipped with algorithms that do statistical analysis to provide some kind of measure of risk. There is much variation as to how this is accomplished. Users should be aware that for commercially developed models all such analyses attempt to measure only those project risks that are related to model parameter settings. In most major projects, these are only a subset, sometimes a minor one, of the total risk picture. Some models designed solely for risk analysis are much more comprehensive in this regard than complex parametric models designed primarily for cost estimating.

1.1.2.4 Using the Results

The basic use of parametric model output is the estimation of a range of cost (see Appendix H), or an analog of that cost, such as labor hours or project duration. But the estimation of cost can have multiple purposes. Among them are:

- **Proposal pricing.** The most rigorous use is to support determination of a bid amount for a desired project contract. A careful audit of the entire estimating process is likely.
- **Rough estimates.** A less rigorous use is to obtain a rough estimate of cost, often for purposes of determining affordability, or competitive advantage.
- **Trade studies.** Trade studies examine cost effects of different product design options and/or the cost effects of different ways of organizing the project.
- **Active project management.** A use that is becoming more common is to assist in active management of the project. To do this, the complex parametric model is coupled with a process called earned value management (EVM). EVM measures useful work accomplished versus resources expended to determine if the project is meeting expectations. A parametric model can make forecasts of costs of work remaining. These results can be helpful to project management in detecting the need and scope of corrective action.
- **Sanity checks and cost realism.** If a project has decided to use another estimating method as the primary method for a proposal, it may nevertheless want to use a parametric backup estimate as a “sanity check.”
- **Competitive analysis.** In competitive situations, it is often desirable to try to assess a competitor’s costs. One useful approach for doing this is to collect parametric information about the competitor and input into a parametric model. If there is some uncertainty about the parametric information, use of a model with risk estimating features can be helpful.

1.2 Cost Estimating Relationships

A cost estimating relationship (CER) is the foundation of the art and science of estimating resource needs in projects using parametric methods. The parametric method comprises collection of historical cost data and reducing it to mathematical forms that can be used to estimate similar activities in future projects. The mathematical forms are called CERs. They are most commonly algebraic equations, but sometimes they are tabulated data.

In this section, we discuss certain aspects of CERs, namely:

- Data characteristics and sources;
- Data normalization;
- CER development;
- CER validation.

CERs are mathematical in nature, and the math can be somewhat advanced, but our discussion of the mathematics in this section will be at a rather elementary and cursory level. See Appendix B for a deeper exploration of the mathematics.

1.2.1 Data Characteristics and Sources

CERs are created for the main purpose of being able to quickly and reliably estimate future project costs. For that reason, we must be clear about what a CER is supposed to estimate. A key issue is comparability of the historical data and the future costs to be estimated. Comparability has many aspects. Several of the more important ones are discussed in this section. Also discussed are commonly used data sources and the importance of data context.

1.2.1.1 Comparability of Activities

Ways of comparing project activities are almost limitless. Our interest in comparison will be limited to examination of project activities that can create a material difference in cost. Even so, it is impossible in the scope of this handbook to capture all that can be imagined. It is the job of the CER builder to be sure that he or she has captured all significant project activities that matter.

Some project activities have shown by experience to cause material differences in cost; they also frequently occur. This includes:

- Timing;
- Labor versus material;
- Recurring versus non-recurring;
- Overhead versus direct;
- Production quantity and rate;
- Team skills;

- Team tools;
- Volatility;
- Accounting changes;
- Special constraints.

Timing

Timing of historical versus future costs is of importance for at least one main reason: variations in the value of currencies, also called inflation or deflation. Adjustments must be made due to these changes in the value of currencies.

Another consideration is the number of years in the period of performance and total cumulative data from inception to completion.

Labor versus Material

In common parlance the difference between labor and material is clear. But it is not always clear in the world of accounting, which is the source of the data used to build CERs. Major integrating contractors commonly refer to anything they buy as material, regardless of its labor content at the source. Lower level contractors may do the same thing, but their labor content may be considerably different than an integrating contractor's.

Recurring versus Non-recurring

Costs related to initial development of a product are frequently referred to as non-recurring costs on grounds that they will only occur once. Costs related to production of a product are referred to as recurring costs on grounds that they will recur every time the product is built. Hardware projects commonly have both kinds of costs, while software projects commonly have only non-recurring costs.

As long as the definitions of what is included in each category remain consistent, there are no problems. But different organizations have been known to adopt different accounting practices in this regard. For example, organizations typically treat engineering design effort as non-recurring costs, while a few bundle the engineering effort into an overhead account and apply it as a burden to production labor costs. The difference in production burdened labor rates is substantial. See Appendix I for additional discussion of recurring and non-recurring costs.

Overhead versus Direct

As an accounting convenience, accountants often make a distinction between overhead costs and direct costs. Generally speaking, costs closely related to the purpose of individual projects are classified as direct, while costs more remote from the purpose of individual projects are classified as overhead. Unfortunately, accountants have no uniform standard for making this distinction.

The usual manner of recovering direct costs is to list them explicitly in contract proposals. The usual manner of capturing overhead costs is to bundle them and apply them as burdens to selected labor rates or sometimes to material purchase costs. While this to some extent masks the true costs of activities, it is not particularly troublesome to CER builders as long as the allocation percentages remain about the same. Unfortunately, different organizations have different overhead structures, so mixing data from different organizations can be troublesome. Moreover, even within one organization, overhead rates and thus burdened labor rates, can sometimes vary as often as daily.

Production Quantity and Rate

If cost data rolls up the cost of many items produced, as opposed to separately enumerating the cost of each item, the data is not useful unless the quantity produced is known. Of lesser effect on data usefulness is the rate of production, the effects of which are more subtle and variable.

While quantity is the main driver of total production cost, the well known learning effect can also have a considerable impact.

Team Skills

In some organizations team skills are relatively constant and therefore of not much concern to CER builders. However, the modern trend is for competitive project organizations to engage is some form of continuous improvement, thereby becoming more cost effective in their work. This takes various forms, such as CMMI, and various ISO classifications, but team self-improvement is the common purpose.

Team Tools

Quality teams cannot be their most effective if they work with poor tools. “Tools” can include everything from buildings to production machines to computers and software. As tools improve, cost effectiveness increases.

Volatility

The most cost effective project environment is one in which project requirements, labor force, and infrastructure is stable. Volatility in any of these factors can increase costs.

Accounting Changes

Companies change their accounting systems for many reasons. There are mandated changes from the Government, internal decisions to change cost accumulation procedures, adjustments to account for new ways of doing business, and mergers and acquisitions. In any event, expect to have to reconcile history to account for these planned changes.

Special Constraints

Various kinds of special constraints can seriously affect cost. Among them are:

- Overly short or long project schedules;

- Staff shortages, especially of skilled people early in the project;
- Ill advised attempts to reduce costs;
- High levels of project secrecy.

1.2.1.2 Commonly Used Data Sources

A data source commonly used by CER builders is an organization's formal and official books of account, often referred to as the general ledger. Other sources, sometimes of lower reliability, but often more detailed, include:

- Engineering design records;
- Program reviews (PDR, CDR, and so forth);
- Manufacturing records;
- Departmental records;
- Purchase orders;
- Cost reports to customers and others;
- Special cost studies;
- Industry surveys;
- Government reports;
- Cost proposals.

The latter should be used only as a last resort. Estimates based on proposals are often viewed with suspicion.

1.2.1.3 Importance of Data Context

The commonly used sources listed in the previous section are sometimes big on numbers but weak on detailed descriptions. The minimum detail needed by a CER builder is the cost's accounting characterization, but that is scant information and can sometimes be confusing. Other information that's highly useful, and sometimes obtainable only from informed individuals, includes:

- Production quantity;
- Production rate;
- Project schedule;
- Nature of the product;
- Major project perturbations, especially changes in requirements;
- Make versus buy content;
- Special problems or situations encountered, such as introduction of a new technology, major changes in the available skill mix, and so forth.

Context information such as the above is useful to the CER builder in determining whether or not the future project to be estimated is sufficiently similar to the historical projects from which data was taken.

1.2.2 Data Normalization

Data normalization is a process whereby a CER builder attempts to correct for dissimilarities in historical data by putting the data into uniform format. In principle, if all of the historical data comes from the same organization, if that organization is not changing or learning, if the organization is repetitively doing the same kind of work with no improvements in efficiency or technology, and if the national currency does not fluctuate in value, historical data would not need to be normalized.

For historical data, normalization is virtually always necessary. How much is necessary depends on what is changing and how fast it is changing. It also depends on how much accuracy is needed in the CER being built.

The ability of a CER builder to do normalization is almost always subject to the limitation caused by unrecorded data. Not everything that affects cost in a historical project is recorded in the official books of account. If a CER builder is fortunate, a historical project will have recorded certain vital contextual information in some kind of anecdotal project history.

While some normalization adjustments are of the types that require an initial injection of expert opinion to get the ball rolling, others are more mechanical. The two that are most nearly mechanical are adjustments for inflation and production quantity.

To normalize for inflation, the CER builder will locate a prior year's inflation table appropriate to a given product or product mix. Using this table, the historical costs will be adjusted to a common desired base year. The CER will then make its estimates based on the new base year currency values. If the CER user wishes the cost output to be in a different base year currency, he or she must use a table of inflation that includes that base year. If the new base year is in the future, as is likely, the inflation values given by tables will be estimates made by Government or industry economists.

If every unit produced had the same labor hours and material cost, or deviated only slightly from the average of those values, adjusting for differences in production quantity would simply be a matter of dividing the historical total cost by its production quantity to get an average value. Unfortunately, in most cases this simple linear adjustment will usually be far off the mark if the production quantity is more than three or four. In production where the quantity is more than just a few, a highly non-linear phenomenon known historically as "learning" commonly takes place.[2]

[2] Little or no learning may take place in highly automated factories. However, there may be improvements due to better methods and equipment.

Sometimes this phenomenon is called cost improvement or other names, but the basic idea is that the work force gradually improves its ability to build the product. This involves both human understanding and motor skills. "Learning" can also be due to investments in better tools and processes. "Learning" may affect material purchase costs through better buying practices, reduction of manufacturing scrap, and measures to reduce various material attrition losses (e.g., due to rough handling, theft, damage in transit, and so forth).

The learning process has been found to be non-linear in that it affects every item built somewhat differently. Without delving too deeply into the mathematics involved, we can say that the learning process appears to be best represented by a non-linear equation of the form:

$$y = ax^b$$

The above equation will plot as a straight line on log-log graph paper.

In one theory of learning, called the unit theory, y is the labor hours or cost associated with unit number x in the production sequence, a is the labor hours or cost associated with the first unit produced, and b is called the natural learning slope. The value of b is almost always negative, reflecting the fact that unit costs decrease as production quantity increases.

In the other theory, called the cumulative average theory, y represents the cumulative average cost of units 1 through x, a again represents the first unit cost, and again b is the natural learning slope.

Learning slope is commonly given as a percentage. The percentage expression of learning is related to b through an equation that can be found in Appendix B.

1.2.3 CER Development

The basic idea in CER development is to 1) identify one or more parameters of a product or project that best explain its cost, 2) find some historical data that are representative of the desired cost, and appropriately normalize it, and finally, 3) identify one or more mathematical functions that "fit" the data and that can be used to estimate future costs based on similar plans about future projects.

The world of useful mathematical functions is extensive. However, most cost data sets arising in practice have fairly simple shapes. This allows good fits using relatively simple functions. The functions used are mostly the polynomials of orders 1 and 2, the power law, the exponential function, the logarithmic function, and some variations on these.

The most elementary function commonly used in fitting to data is the polynomial of order 1, also known as the straight line. If a scatter plot of data appears to be compatible with a straight line, then the function to be fitted would be the equation of a straight line, namely:

$$y = ax + b$$

Where a and b are constants. Figure 1.5 is such a scatter plot.

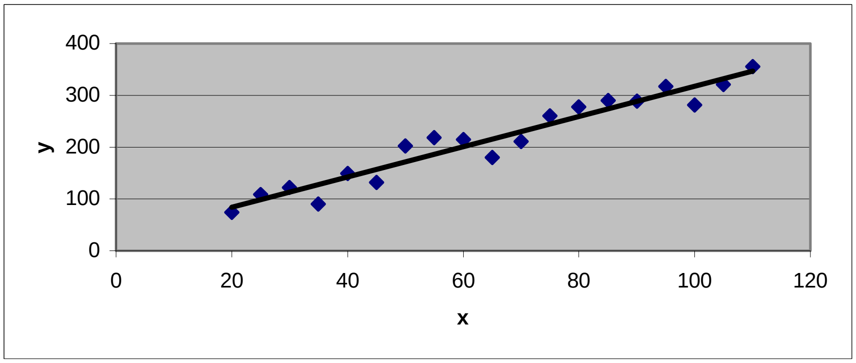


Figure 1.5 Typical Linear Scatter Plot

In the 19th century the famous mathematician Carl Friedrich Gauss (with contributions from others) developed a process called least squares, also called ordinary least squares (OLS) for obtaining an optimal fit of data to a polynomial of order 1 or higher (e.g., a straight line, a quadratic, a cubic, and so forth.). The fit is optimal in the sense that the sum of the squares of the fit errors (known as "residuals") is minimized. See Figure 1.6 below. Hence the name "least squares."

Certain so-called "transcendental" curves favored by parametric analysts, such as the power law, the exponential, and the logarithmic, cannot be fitted directly by the OLS process. The technical reason for this is that they are not linear in their coefficients. However, the most useful ones can be converted to a linear form by a certain mathematical transformation, and in that form the OLS process can be used.

One problem with OLS, bothersome to analysts, is the nature of the error term of the resultant fitted equation. Using a simple linear equation to represent the class of polynomials, the equation including error term can be written:

$$y = a + bx + \varepsilon$$

Where ε is the error. Note that the error term is additive, which is not always appropriate. When certain non-linear functions are converted to linear form so that OLS can be performed, the error term is multiplicative, also not always appropriate, but usually of more interest in the cost estimating context:

$$y = (a + bx)\varepsilon$$

A graphical illustration of the difference between additive and multiplicative errors is presented below in Figure 1.6.

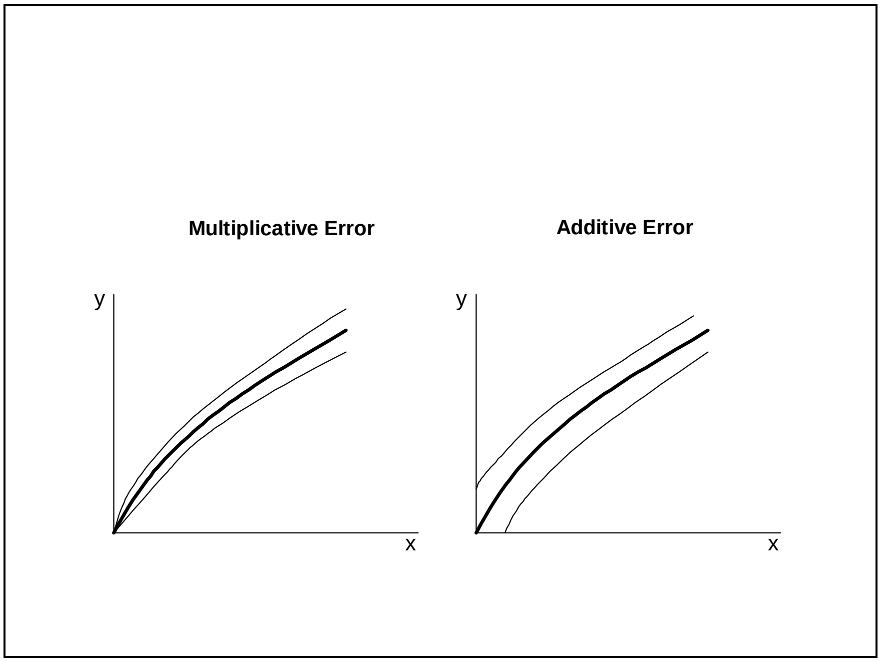


Figure 1.6 Comparison of Additive and Multiplicative Errors

To the problem of error inconsistency can be added the inability of OLS to fit certain interesting and sometimes useful non-linear functions that cannot be transformed to a linear equivalent, such as $y = ax^b + c$.

Various fixes have been proposed for these problems, but probably the most popular is called the General Error Regression Model (GERM). With GERM, one can choose to use either an additive or a multiplicative error approach, and one can fit virtually any function one chooses to fit to the data, however non-linear it may be. Today, GERM is widely used, and it can be implemented on most computer spreadsheets.

1.2.4 CER Validation

Once we have gone through all of the steps of finding and normalizing data and fitting one or more functions to it, we naturally want to know if the result is any good. The ultimate test of the goodness of any CER is whether or not it can predict project costs with reasonable accuracy. Unfortunately, we can never know that for sure until we estimate the costs, do the project, and compare the results to what the CER predicted. But by then, if we had a weak or inaccurate CER, it would be pretty late to find that out, and damage could have been done.

So, a lot of effort is typically expended on CER validation before a CER is used for any risky purpose. Validation activities are typically 1) practical, 2) mathematical, and 3) judgmental. The most practical thing that can be done is to use the CER to estimate one or more projects that have already been completed and see if the answer is accurate to within expectations.

Several mathematical tests are available for CERs. We will briefly discuss three of them:

- Standard error of estimate (SEE);
- Average percentage bias;
- Coefficient of variance (R^2).

Standard error of estimate (SEE). SEE is the root mean square (RMS) value of all percentage errors made in estimating points of the data. It is similar in nature to the more well known standard deviation (σ) statistic. SEE measures how well the model represents its own underlying data, given the scatter.

Average percentage bias. This is the algebraic sum of all percentage errors made in estimating points of the data averaged over the number of points. Bias measures how well percentage over and underestimates are balanced.

Coefficient of Variance. This statistic, written as R^2, is undoubtedly the most commonly used measure of goodness of fit, although many say it is not the best. It measures the amount of correlation between estimates and corresponding database values, that is, the degree of linearity between two quantities.

CERs have to be sanity checked. These checks can take various forms from management reviews to in-depth audits. A growing practice is to form an integrated product team (IPT) to review all of the steps of CER creation with a view to assessing their validity. The activities of these IPTs can resemble "murder boards," in that they attempt to punch holes in all validity arguments. A CER that survives such a process is likely to be of high quality.

1.3 Complex Models

What is a "complex" parametric tool or model, and how does it differ from a cost estimating relationship (CER)?

In this section, we will try to make those differences clear to help readers better understand how complex tools are built and how they fit into the estimating process.

1.3.1 Comparison of Complex Models and CERs

"You can't estimate the cost of something if you don't know what it is." The fact is, if you have some knowledge about something, you may be able to make an approximate estimate for it, but the more you know, the more accurate your estimate is likely to be.

If knowing more can lead to a more accurate estimate, then it should follow that a model that asks for more information, every thing else being equal, will generally give you a better result than a model that asks for less, assuming that you can provide the extra information. So, as the primary descriptor of "complex" models, we can say that they are models that ask the user for more information than ordinary CERs do, typically much more. Initially there is limited technical description upon which to build a model. As more technical detail is established, the model expands becoming more complex. Thus an early version of an in-house built cost model will be simpler, requiring less input, than a mature detailed model requiring much more input.

A typical CER will ask for one, two, three, or perhaps four pieces of parametric information. A complex model, by contrast, may ask the user for 20 to 40 pieces of information. Its construction may involve a variety of statistical analyses and inferences, generally not limited to regression, and will inevitably use expert judgment about the way the world works in order to arrive at some of its results.

A complex model asks for more information than a single CER, and generally it also provides more information in its output. A typical CER will give one answer, usually a single point estimate of total cost, labor hours, material cost, time, weight, and so forth. A complex model might provide all of that, and a great deal more. For example, a complex model might provide a range of costs that includes risk and uncertainty, information about project team skill mix and size, spend profile, activity scheduling, facilities required, and so on.

Because a complex model is typically designed to report many kinds of information, it tends to be algorithmically robust compared to a CER. While a CER may comprise as little as one algebraic equation, a complex model could have dozens of interactive equations, as well as look up tables, if-then logic ladders, and even iterative procedures such as non-linear equation solving or Monte Carlo simulation.

Mathematics is a powerful tool, but unaided, it is far from capable of putting together the analytical approach of the typical complex model. Considerable expert intervention is required as well. While a reasonably competent mathematician or statistician may be able to build a valid CER given a regression tool and a set of fairly clean data, that person would probably be unable to build a valid complex model. Why? Because of inexperience with the objects for which the estimates are needed, and perhaps also because of inexperience with the way those objects are used, come into existence, or even what they look like. This is why a good parametrician has a technical background, versed in finance with strong math and statistical skills.

Another difference between CERs and complex models is that CERs often have much wider scope. There have been CERs (also known as "rules of thumb" in this case) that proclaim that it takes x dollars to get a pound into space. A complex model is unlikely to make such a simplistic claim making its versatility limited to well defined programs.

1.3.2 Complex Tools in the Estimating Process

Early estimates of a new project, especially a project involving one or more new concepts, are far more likely to be too low than too high. The reason is lack of knowledge about the true scope of the project, and the obstacles that need to be overcome, otherwise known as the "unknown unknowns."

Many projects today are extensive in scope, cost, and duration. Such projects typically operate under fairly firm cost constraints. These constraints often have names such as target cost, design-to-cost (DTC), life cycle cost (LCC), total ownership cost (TOC), or cost as independent variable (CAIV). Where do such cost constraints come from? How are they set? Often, they come from complex cost models. Many of these models have the capability to not only provide a most likely cost, but also provide costs at various probabilities of success. Careful use of such models can help establish not only total cost targets, but also lower level targets that are challenging but doable with reasonable effort.

Unless a project involves totally familiar effort, the stakeholders in a consequential project will want to examine different ways of executing it. Such examinations are called trade studies. They can range from macro level studies conducted by the customer or project management, such as which development team to use, which factory to use, what kinds of tests to conduct, and so on, to micro level studies conducted by engineers, such as which wing shape is best, which fuel control valve is best, and so forth. Seldom can even a complex model embrace all of the considerations involved in such choices, but they can often be very helpful with the comparison of the resources required aspects of a trade study.

Stakeholders will be interested in seeing if a project is meeting expectations, and if not, what liabilities they are assuming by continuing. Earned value management (EVM) is a much touted tool for this purposes, but unaided, it has limited predictive power. A complex estimating tool can enhance the effectiveness of EVM by providing reasonably accurate predictions of cost outcomes given the current situation. A complex tool can vastly aid decisions about whether to continue, and if continuing, how much unplanned time and money will be needed.

Complex tools can also assist pricing, production quantity, and marketing decisions. Pricing below estimated cost is not feasible in the long run, but can be an effective short run strategy. But to safely develop such a strategy, you must be pretty sure what the cost is. Production quantity is a key decision variable, being influenced by funds availability, production costs, and market demand. See Chapter 8 for an additional discussion.

1.3.3 Development of Complex Tools

The development of complex tools will be considered under three sub-headings:

- Data and cost drivers;
- Mathematics and logic;
- Outputs.

1.3.3.1 Data and Cost Drivers

Parametric estimating, like all other estimating, is based on analogy. To estimate the cost of anything, we need information about what similar things have cost in the past. We call such data “historical costs.”

Complex models almost invariably contain substantial input of expert judgment. But even the best experts can misjudge on occasion. Therefore, expert judgment must be subjected to as much validation as reasonably possible.

In an in-house complex model, there is likely to be both more data and/or possibly better expert judgment because of direct experience with the products. But, most model builders often suffer from shortages of data, and commercial model builders are no exception. For certain types of exotic hardware, there may be only a small amount of data in the whole world, and most of it may be hidden from view for competitive reasons. Because of that, developers of commercial models generally rely more on expert judgment than in-house modelers.

Two different organizations challenged to build an item will have different costs, sometimes substantially different. Which organization's costs do you try to capture in a parametric model intended for sale to many different organizations? The usual answer is that a commercial complex model will generate costs that are "typical" of the industry. The model builders may not know exactly who will use their model, and they almost certainly cannot be sure whether their users are above or below industry norms, so they will try for what is typical. Depending on the data they have and how they process it, "typical" may be closest to mean, or median, or mode. This creates the need to calibrate commercial complex models. See Section 1.1.1.5.

The process of interpreting data for use in a model is commonly called normalization. There are some guiding principles that should be followed. See Section 1.2.2 and Chapter 2 for more discussion on this topic.

Once data is normalized, the remainder of the development process is highly dependent on the outputs the developer wants the model to produce. The most common outputs of commercial models are:

- Development cost;
- Development labor hours;
- Development material cost;
- Production cost;
- Production labor hours;
- Production material cost.

Commonly, all of these results are produced in hardware models, but only the first in software models. In software development, the "production" activity called "coding" is generally regarded as part of development and thus defined by convention as non-recurring.

Until recent years, the universally accepted primary driver for software development cost was the number of lines of delivered working code. That primacy still exists in many companies, but others have come to prefer other metrics. Of those, the most widely accepted appears to be counts of function points. The function point method counts not the lines of code, but the number of distinct functions that the software must perform. There are several ISO standards for counting application functionality, defining the rules for counting and weights to be used in computing a summary function point count. Other primary metrics have been considered, such as number of "use cases," but these as yet have not had wide acceptance.

There is increasing interest in non-weight primary drivers for non-electronic hardware. One reason is that in general, all that engineers want to do with weight is decrease it, and doing that can cost more than leaving it alone. So, it is not a particularly useful metric in trade studies that consider cost, as most do. Many future models will have as primary drivers relatively difficult or expensive to meet requirements or product features. For example, for a space based telescope, the primary drivers might be diameter, number of imaging elements, number of non-imaging elements, and optical quality.

Lower level drivers for both hardware and software models are generally of one of three types:

- Product operating environment;
- Project team composition;
- Project team environment.

Parameters often considered under product operating environment for hardware include:

- Type of service (frequently called "platform");
- Functionality (e.g., provide lift, radiate RF frequencies, and so forth);
- High operating pressures;
- High margins of safety;
- High levels of reliability;
- Level of radiation experienced.

Often considered for software:

- Execution time constraints;
- Memory constraints;
- Machine volatility;
- Machine of use differences from development environment.

Parameters often considered for project team composition include:

- Average team member capability;
- Average team member experience;
- Cohesiveness of team members.

In addition, experience with the programming language is often considered in software development.

Project team environment typically includes such parameters as:

- Availability of modern tools;
- Knowledge of how to use available tools;
- Degree of geographical dispersion of team members;
- Volatility of the design requirements;
- High level of security requirements.

Cost drivers are not all equal. Some are more sensitive and cause larger cost variations than others. In an ideal complex model, all cost drivers would have similar effect, but that ideal is unattainable. The reality is that some drivers will be "big" and others will be "small." Users should be aware of this and should know which is which. Model builders tend to make this information readily available.

1.3.3.2 Scope of Estimates

Generally, complex models focus on estimating development and production cost. Some go beyond this and deal with what is commonly known as "operations and support," or the costs of operating and maintaining fielded systems.

System integration, as opposed to system<u>s</u> integration is another important scope issue. All complex models deal with one or more <u>systems</u>, in some sense of that word, and many deal with systems within a system. Recently, because of growth in what technology is capable of doing, we hear more and more about systems of systems, or mega-systems. Commercial model builders continually try to upgrade to enable users to better deal with these higher level integration issues. Users must be aware of what level of integration complexity their complex model is capable of dealing with, and limit their expectations accordingly. The practitioner should be sensitive to this and include factors for these costs in the in-house developed model.

1.3.3.3 Mathematics and Logic

There exists a large diversity of mathematics and logic used in complex models. Here is a list of some major categories and a few examples:

- CERs;
- Algorithms.

CERs

Virtually all CERs are empirical fits to data. The most commonly used tool for fitting is linear least squares regression, but other tools are often used, for example:

- Non-linear least squares;
- Gauss-Newton.

A variety of curves is used. The choice of curve is most often on the basis of some criterion of "best fit." The following list includes the most commonly used curves, in no particular order:

- Linear.
- Segmented linear;
- Polynomials of various degrees;
- Power law;
- Log normal;
- Exponential;
- Rational;
- Various statistical distributions;
- Power series.

Some simple CERs have only a single explanatory variable (cost driver). But in complex models there are often two or more. Most complex models, especially custom in-house developed models use a breakdown of hardware or software into a WBS. Many of the core CERs are developed on lower level data, which may be more readily available.

Algorithms

Complex models use a multitude of algorithms for a variety of purposes. Among the most commonly seen are the following, in no particular order:

- **Phasing and spreads.** Some complex models strive to provide some notion of how costs will be spread. Some limit this to an annual presentation, but others show costs at the monthly level.
- **Inflation.** Closely related to phasing and spreads is the notion of effects of inflation. Typically, these come from economic forecasts made by Governments or by corporate or academic economists.
- **Allocations.** Most complex models strive to allocate costs to various subcategories of labor skills or typical project activities. The percentages allocated to each subcategory are usually built into the model. Sometimes the user is allowed to modify them as a form of calibration.
- **Monte Carlo simulation.** The iterative statistical sampling tool known as Monte Carlo simulation is most frequently used to generate depictions of risk or uncertainty called the range estimate or probability distribution. It can be used as an aid to estimating costs, as for example, in finding the most logical cost outcomes based on primitive data.

- **Scheduling.** The attention paid to scheduling seems to vary widely among complex models. In part, this is probably because cost is often the first interest of model users. But for some users, scheduling is of great importance because of high interest in timing of cost demands versus timing of availability of funds. A few models provide elaborate scheduling capability using critical path networking techniques. Others limit scheduling to user defined or parametrically estimated start and finish dates for major activities. Still others may estimate some duration values but otherwise have little or no scheduling capability.
- **Learning.** In production operations where more than a very few items will be built, getting an accurate estimate can require careful consideration of the learning effect. Complex models usually give the user a choice of one of two widely accepted theories of learning, the unit theory and the cumulative average theory. Learning slope is usually set by the user, and various alternative situations may be provided for, such as prior learning, breaks in learning, segmented learning slope, and so forth.
- **Earned value management.** Earned value management (EVM) is a much touted tool for tracking whether or not a project is meeting cost and schedule expectations. It can work even better when it is built into a complex model, and some models include it.
- **Special adjustments.** Complex models typically require a variety of special adjustments to accommodate user needs. For example, consider the situation when a project undertakes to design and build a piece of hardware that is similar to, yet in some ways different from, hardware it has built before. The heritage from the previous work will tend to reduce the cost of development in the new project. Therefore the model builder must create an algorithm that adjusts the cost based on the user's description of the amount of heritage.
- **User error detection**. On occasion, a model user may create inputs that are inconsistent, or that exceed the design range of the model. If not warned, the unaware user could get a result that is grossly in error. A common practice is to provide user warning notices.

1.3.3.4 I/O Management

Model builders must facilitate rapid input as much as possible. Complex models also generally provide many outputs, most of which are unwanted at any point in time. Model builders must make it possible for the user to quickly isolate the needed outputs.

In recent years, users have tended to link complex cost models with other complex models, to make even more rapid the transfer of cost driving information. This sometimes results in the cost model being semi-automated.

Input Management

Complex models may require a variety of inputs. Here are perhaps the most commonly seen ones:

- **List of items to be estimated.** The most common formulation of such lists is the cost or work breakdown structure. This is typically a multilevel hierarchical list that places the items to be estimated at the lowest level of the hierarchy.
- **Presets or scenarios.** Various schemes are in use to simplify the work of the user. These usually take the form of some method of quickly indicating a certain condition or conditions that apply to the item to be estimated. The preset or scenario results in automatically setting certain parameters to values that are typical of that condition. The model user has the option of revising these if they are deemed to be unsatisfactory. An example of a preset could be to specify that the item to be estimated is to be a part of a manned aircraft. That could result, for example, in a parameter called Product Quality being automatically set at a high level, thus raising the cost above what it would be if the product was to be used in an automobile.
- **Input importing.** Parametricians very often make use of information that is already stored in electronic form elsewhere than in the complex tool itself. User friendly complex tools provide means for easy and rapid import of such information, especially from spreadsheets. A popular import capability is to build the WBS in a spreadsheet, then bring it into the complex tool. This tends to simplify WBS editing. Values of parameters are also popular imports.
- **Product parameter inputs.** These are inputs that describe features of the product. For hardware, the most common is weight. Others typically relate to design features, requirements, quality, and complexity of the product. For software, the most common is lines of code and language. Other measures of software size include function points and use cases. Other frequent software product descriptors have to do with reliability, complexity, and team environment and characteristics.
- **Product operating environment parameter inputs.** Products are generally designed for use in a particular environment, such as fixed ground, mobile ground, sea, manned aircraft, and so forth. Sometimes the parameter is about the general operating environment, and sometimes it is about specific environmental requirements, such as high internal pressure, exposure to space vacuum, high temperature, radiation, and so forth. The software environment includes speed and memory constraints, and the operating system.
- **Project team composition parameter inputs.** Some project teams are excellent, and some are less than excellent. Team capability and experience can make a large difference in project cost.

- **Project team environment parameter inputs.** Teams functions better in an ideal environment. That includes close proximity of team members, lack of distracting situations, and the right tools.

Output Management

Complex models typically provide a variety of reports and charts for users. More detailed charts typically present detailed breakouts split in a number of ways. Pie charts and various other graphical formats are commonly seen.

Most complex models provide means for the user to design his or her own output reports. These custom reporting utilities generally allow the user to select just about any combination of dozens of available outputs, and inputs as well, and form them into a custom report. Typically, custom report formats can be stored and used repeatedly. Most complex models provide for output export to other applications.

Integration of Cost Models with Other Models

More and more, product development activities are being streamlined to decrease the time necessary to provide product availability to end users. To that end, collocated development teams have been formed in some organizations, with each team member positioned at a computer that is linked to all of the other computers in use by the team.

A cost analyst is generally a member of the team, and it will be his or her function to keep track of the costs that result from the designs, versus the goal costs, and also to be a kind of cost referee in tradeoffs where cost is a dimension of the trade space. To have the power to deal with the many complex cost issues that are likely to arise, the cost analyst in these teams is generally equipped with one or more complex, general purpose models, and perhaps several simpler models or CERs designed for specific types of estimating.

In this development environment, manual transfer of cost driving technical data to the cost model, or models, slows the activity down considerably, and can be a cause of error. Because of this, most development teams today highly favor semi or fully (where achievable) automated transfer of technical information to the cost model. Much of the focus of model building today is to facilitate rapid data transfer.

1.3.4 How Good Is the Estimate?

Project managers and others frequently challenge results from parametric models, especially when 1) they appear to run counter to what they hoped the result would be, or 2) when their subordinates insist that the results are too high or two low. It can well be that parametric results are “wrong.” Wrong inputs will almost certainly give wrong outputs. But will “right” inputs always give right outputs? That is a complicated question that we examine in this section.

1.3.4.1 Importance of Builder and User Skills

Every method of estimating that you can imagine has some kind of "dials" that you can "spin," giving you "any answer that you want." True, it may be physically and mentally easier to spin them on a parametric model than with some other methods, but with any method, you can only get the right answer when you are a seeker after the truth. A qualified parametrician does not spin the dials. He or she thoughtfully strives to find the best setting for each dial.

While ease of spinning the dials can lead to bad results for parametricians not rigorously seeking the truth, it is one of the great advantages of parametric estimating. A qualified parametrician, using a model with which he or she is well familiar can produce in a few days a major project estimate that would take a month or six weeks to do by collecting and integrating bottom-up estimates from project team members. Moreover, if the project is not yet well understood, the parametric estimate is likely to be more accurate.

A complex model contains virtual intelligence (VI), also confusingly called artificial intelligence (AI).

VI does not come out of a math textbook or a course in statistics. It comes from some understanding of the nature of the object to be estimated (what it is). Not only what it is, but how it comes into existence, what function it fulfills, and the environment in which it will be used.

Other VI in a good complex model has to do with the skills of the parametrician. The point is, the outcome of a project is dependent on the skills of the project team, and the quality of a complex model will depend on the experience and skills of the model builders.

1.3.4.2 The "Black Box" Criticism

A few complex models are "open." That means that all algorithms and perhaps even the foundational data are open to users for inspection. An excellent example is the COCOMO II software estimating model developed by Dr. Barry Boehm and described in minute detail in his pioneering book *Software Engineering Economics*.

Commercial complex models on the other hand are customarily proprietary. That is, materials furnished to the user describe in detail how to use the model, but they expose neither the algorithms used, nor the exact data sources. For that reason, they are frequently criticized as being "black boxes," that is, models whose inner working are unknown to the user and therefore subject to suspicion.

Although inquiry to the model developer can sometimes result in some degree of explanations, developers generally prefer to maintain confidentiality of most of what they do, for competitive reasons.

So, why should you trust a commercial complex model? Because:

- You fully understand the intended uses of the model, and you have one or more applications that correspond to some of those uses;

- You are willing to have yourself and/or at least one associate professionally trained in the use of the model before you do any serious estimating with it;
- You have investigated and know that other organizations who do estimating similar to yours are using it and are satisfied with the results;
- You use the model at least once to estimate one of your previous projects where the results are known, and you are satisfied with the comparison.

If you do these things and are happy with the outcomes, the black box syndrome is not a concern.

1.3.4.3 What the Estimate Means

As noted earlier in this section, commercial model builders seldom know exactly who will use their model, and exactly how they will use it. Therefore they will strive to make the model produce what they believe to be "typical" results. Unfortunately, what they think is typical may appear atypical to some users. Companies do not all do business the same way.

The model results also reflect various cost sensitivities as perceived by the user. Those sensitivities may be somewhat different in your organization. Common differences are the fairly large variances in overall cost results, variances in ratios of materials to labor, and the allocations of labor skills.

Such differences often appear in early attempts to use the model. Model builders fully realize this phenomenon, and for that reason urge users to take the time and trouble to do at least one substantiation or calibration. We discuss calibration in the next section.

1.3.4.4 Calibration to Improve the Estimate

Some commercial complex models have features for calibrating the model to account for differences in the way of doing business.

1.3.4.5 Validation for Customers

When a bidding situation is sole source or when the project funding is "cost plus," customers will usually demand a thorough explanation and justification of cost estimates. When a bidder uses a commercial complex model to prepare an estimate, the customer will naturally have concerns about the validity of the estimate. These concerns basically boil down to concerns about the validity of the model in general, and also about the validity of the model for the particular contractor and project.

Model builders have in the past addressed these concerns by means of limited disclosures of model architecture and algorithms. Bidders have addressed them by full disclosure of parametric inputs and steps taken to calibrate against historical costs.

Many customers today are accustomed to receiving parametrically derived estimates and are not concerned about them if they are properly supported.

1.3.4.6 Validation of Expert Judgment

We mentioned previously that complex models inevitably have some need of expert judgment. Unfortunately, uncontrolled use of "expert judgment" can result in an inaccurate model that eventually will have no credibility. To the extent possible, expert judgment must be validated. With regard to minor design issues in a model, the opinion of one expert may be enough, but in consequential issues, use of a single expert can be risky. It is important that the experts be the best that are available, and the most knowledgeable in the project being estimated.

CHAPTER 2

Data Collection and Analysis

All parametric estimating techniques, including cost estimating relationships (CERs) and complex models, require credible data before they can be used effectively. This chapter discusses the processes needed to collect and analyze the data used in parametric applications, as well as data types, sources, and adjustment techniques. It also:

- Identifies sources of information that can be collected to support data analysis activities;
- Describes various methods of adjusting raw data to put it on a common basis (i.e., data normalization);
- Discusses the importance of collecting historical cost and non-cost (e.g., technical or programmatic) data to support parametric estimating techniques.

2.1 Data Types and Collection

Parametric techniques require the collection of historical cost data (including labor hours) and the associated non-cost data information and factors that describe and strongly influence those costs. Data should be collected and maintained in a manner that provides a complete audit trail with expenditure dates so that costs can be adjusted for inflation. Non-recurring and recurring costs should be separately identified. While there are many formats for collecting data, one commonly used by industry is the work breakdown structure (WBS), which provides for the uniform definition and collection of cost and certain technical information. DoD Handbook *Work Breakdown Structures for Defense Materiel Items* (MIL-HDBK-881A) provides detailed guidance on the use of the WBS. Regardless of the method, a contractor's data collection practices should be consistent with the processes used in estimating, budgeting, and executing the projects from which the data was collected. If this is not the case, the data collection practices should contain procedures for mapping the cost data to the cost elements of the parametric estimating technique(s) which will be used.

The collection point for cost data is generally the company's management information system (MIS), which in most instances contains the general ledger and other accounting data. All cost data used in parametric techniques must be consistent with, and traceable to, the collection point. The data should also be consistent with the company's accounting procedures and generally accepted cost accounting practices.

Technical non-cost data describe the physical, performance, and engineering characteristics of a system, sub-system, or individual item. For example, weight is a common non-cost variable used in CERs and parametric estimating models. Other examples of cost driver variables are horsepower, watts, thrust, and lines of code. A fundamental requirement for the inclusion of a technical non-cost variable in a CER is that it must be a significant predictor of cost. Technical non-cost data come from a variety of sources including the MIS (e.g., materials requirements planning (MRP) or enterprise resource planning (ERP) systems), engineering drawings, engineering specifications, certification documents, interviews with technical personnel, and through direct experience (e.g., weighing an item). Schedule, quantity, equivalent units, and similar information come from industrial engineering, operations departments, program files, or other program intelligence.

Other generally available programmatic information that should be collected relates to the tools and skills of the project team, the working environment, ease of communications, and compression of schedule. Project-to-project variability in these areas can have a significant effect on cost. For instance, working in a secure facility under "need to know" conditions or achieving high levels in various team certification processes can have a major impact on costs.

Once collected, cost data must be adjusted to account for the effect of certain non-cost factors, such as production rate, improvement curve, and inflation – this is data normalization. Relevant program data including development and production schedules, quantities produced, production rates, equivalent units, breaks in production, significant design changes, and anomalies such as strikes, explosions, and other natural disasters are also necessary to fully explain any significant fluctuations in the data. Such historical information can generally be obtained through interviews with knowledgeable program personnel or through examination of program records. Fluctuations may exhibit themselves in a profile of monthly cost accounting data; for example, labor hours may show an unusual "spike" or "depression" in the level of charges. Section 2.3 describes the data analysis and normalization processes.

2.2 Data Sources

The specification of an estimating methodology is an important step in the estimating process. The basic estimating methodologies (analogy, grassroots, standards, quotes, and parametric) are all data-driven. Credible and timely data inputs are required to use any of these methodologies. If data required for a specific approach are not available, then that estimating methodology cannot be used. Because of this, the estimator must identify the best sources for the method to be used.

Figure 2.1 shows basic sources of data and whether they are considered a primary or secondary source of information. When preparing a cost estimate, estimators should consider all credible data sources; whenever feasible, however, primary sources of data have the highest priority of use.

Sources of Data	
Source	**Source Type**
Basic Accounting Records	Primary
Cost Reports	Either (Primary or Secondary)
Historical Databases	Either
Functional Specialist	Either
Technical Databases	Either
Other Information Systems	Either
Contracts	Secondary
Cost Proposals	Secondary

Figure 2.1 Sources of Data

Primary data are obtained from the original source, and considered the best in quality and the most reliable. Secondary data are derived (possibly "sanitized") from primary data, and are not obtained directly from the source. Because of this, they may be of lower overall quality and usefulness. The collection of the data necessary to produce an estimate, and its evaluation for reasonableness, is critical and often time-consuming.

Collected data includes cost, program, technical, and schedule information because these programmatic elements drive those costs. For example, assume the cost of an existing program is available and the engineers of a new program have been asked to relate the cost of the old to the new. If the engineers are not provided with the technical and schedule information that defines the old program, they cannot accurately compare them or answer questions a cost estimator may have about the new program’s costs. The cost analysts and estimators are not solely concerned with cost data – they need to have technical and schedule information to adjust, interpret, and support the cost data being used for estimating purposes. The same is true of programmatic data when it affects costs. As an example, assume that an earlier program performed by a team at CMMI (Capability Maturity Model Integration) level 2 is to be compared to a new program where the team will be at CMMI level 4. The expectation is that the CMMI level 4 team will perform much more efficiently than the level 2 team.

A cost estimator has to know the standard sources of historical cost data. This knowledge comes both from experience and from those people capable of answering key questions. A cost analyst or estimator should constantly search out new sources of data. A new source might keep cost and technical data on some item of importance to the current estimate. Internal contractor information may also include analyses such as private corporate inflation studies, or "market basket" analyses (a market basket examines the price changes in a specified group of products). Such information provides data specific to a company's product line, but which could also be relevant to a general segment of the economy. Such specific analyses would normally be prepared as part of an exercise to benchmark government provided indices, such as the consumer price index, and to compare corporate performance to broader standards.

Some sources of data may be external. This includes databases containing pooled and normalized information from a variety of sources (e.g., other companies, public record information). Although such information can be useful, it may have weaknesses. For example, there could be these types of issues:

- No knowledge of the manufacturing and/or software processes used and how they compare to the current scenario being estimated.
- No knowledge of the procedures (e.g., accounting) used by the other contributors.
- No knowledge on the treatment of anomalies (how they were handled) in the original data.
- The inability to accurately forecast future indices.

Sources of data are almost unlimited, and all relevant information should be considered during data analysis. Figure 2.2 summarizes the key points about data collection, evaluation, and normalization.

Data Collection, Evaluation, and Normalization
Very critical step
Can be time-consuming
Need actual historical cost, schedule, and technical information
Know standard sources
Search out new sources
Capture historical data
Provide sufficient resources

Figure 2.2 Data Collection, Evaluation, and Normalization

2.3 Routine Data Normalization Adjustments

Cost data must be adjusted to eliminate any bias or "unevenness" that are the result of other factors. This is called normalization and is intended to make the data set homogeneous, or consistent. The analyst needs to examine every data set to ensure it is free from the effects of:

- The changing value of the dollar over time;
- Cost improvement as the organization improves its efficiency;
- Various production quantities and rates during the period from which the data were collected.

Non-recurring and recurring costs are also segregated as part of the normalization process.

Figure 2.3 shows the typical data normalization process flow. This does not describe all situations, but does depict the primary activities followed in data normalization.

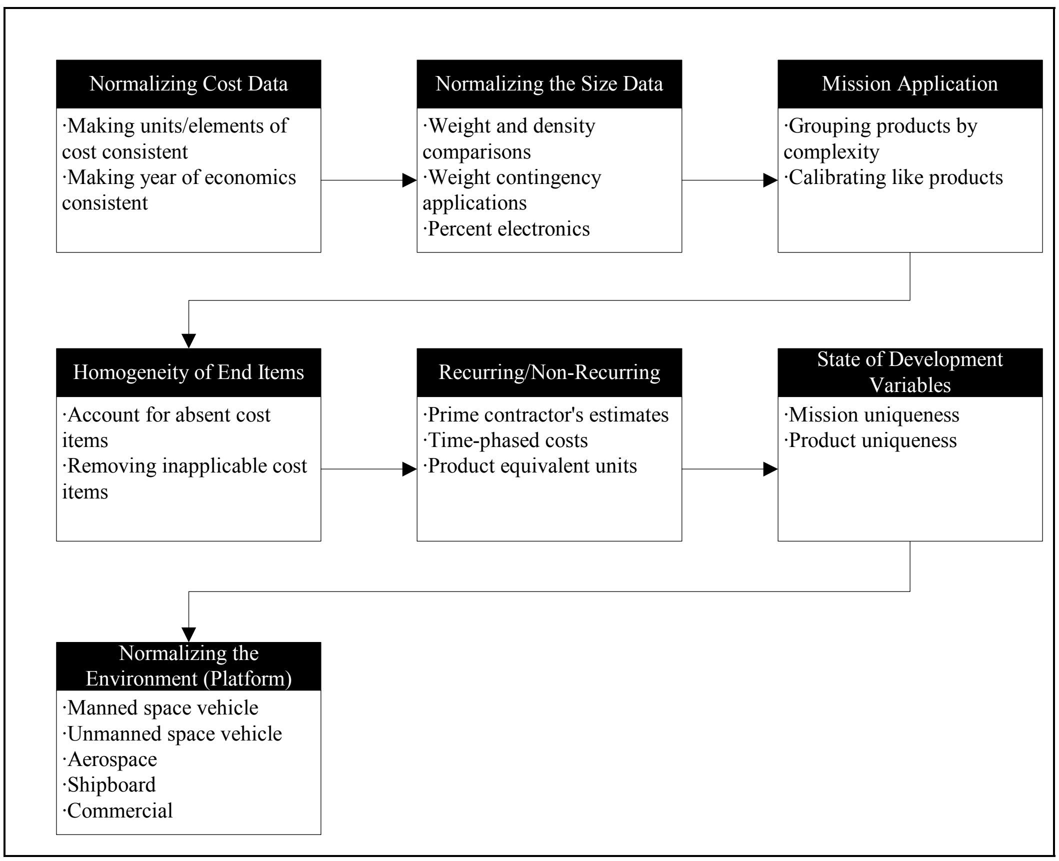


Figure 2.3 Data Normalization Process Flow

Some data adjustments are routine in nature and relate to items such as inflation. These are discussed below. Other adjustments are more complex in nature (e.g., relating to anomalies), and Section 2.4 considers those.

2.3.1 Inflation

Inflation is defined as a rise in the general level of prices, without a rise in output or productivity. There are no fixed ways to establish universal inflation indices (past, present, or future) that fit all possible situations. Inflation indices generally include internal and external information and factors (such as Section 2.2 discusses). Examples of external information are the Consumer Price Index (CPI), Producer Price Index (PPI), and other forecasts of inflation from various econometric models.

While generalized inflation indices may be used, it may also be possible to tailor and negotiate indices used on an individual basis to specific labor rate agreements (e.g., forward pricing rates) and the actual materials used on a project. Inflation indices should be based on the cost of materials and labor on a unit basis (e.g., pieces, pounds, hours), and should not include other considerations such as changes in manpower loading, or the amount of materials used per unit of production.

The key to inflation adjustments is consistency. If cost is adjusted to a fixed reference date for calibration purposes, the same type of inflation index must be used in escalating the cost forward or backwards from the reference date, or to the date of the estimate.

2.3.2 Non-Recurring and Recurring Costs

The prediction of system acquisition costs requires that non-recurring and recurring costs be separately estimated.

Non-recurring costs include all the efforts required to develop and qualify a given item, such as requirements definition/allocation, design, analysis, development, and qualification/verification. Manufacturing and test of development (breadboard and engineering) units, qualification units, and life test units are typically included in the non-recurring cost of hardware end items. Retrofitting and refurbishment of development hardware for requalification is also treated as non-recurring. Virtually all software development and testing costs prior to initiation of routine system operation are non-recurring. Non-recurring integration and test efforts usually end when qualification tests are complete. The non-recurring portions of services costs and some hardware end item costs, such as engineering, are commonly defined as those which take place prior to and during critical design review (CDR). Development, acquisition, production, and checkout of all tooling, ground handling, and support equipment, test equipment, test software, and test procedures are also usually classified as non-recurring.

Recurring costs cover all the efforts required to produce end-item hardware, including manufacturing and test, engineering support for production, and spare units or parts. Recurring integration and test efforts include integration of production units and acceptance testing of the resulting assemblies at all levels. Refurbishment of hardware for use as operational or spare units is usually recurring. Maintenance of test equipment and production support software costs are commonly classified as recurring, while maintenance of system operational software, although recurring in nature, is often considered part of operating and support costs (which may also have non-recurring components). See Appendix I for a more detailed description (from the Space Systems Cost Analysis Group, SSCAG) of recurring and non-recurring cost elements.

2.3.3 Cost Improvement Curve

When first developed, cost improvement was referred to as "learning curve" theory, which states that as the quantity of a production item doubles, the manufacturing hours per unit expended producing it decrease by a constant percentage. The learning curve, as originally conceived, analyzed labor hours over successive production units of a manufactured item, but the theory behind it has now been adapted to account for cost improvement across the organization. Both cost improvement and the traditional learning curve are defined by:

$$Y = AX^{b}$$

Where:

Y = Hours/unit (or constant dollars per unit)

A = First unit hours (or constant dollars per unit)

X = Unit number

b = Slope of the curve related to learning.

There are two interpretations concerning how to apply this equation. In the unit interpretation, Y is the hours or cost of unit X only. In the cumulative average interpretation, Y is the average hours or cost of all units from 1 to X, inclusive.

In parametric models, the learning curve is often used to analyze the direct cost of successively manufactured units. Direct cost equals the cost of both touch labor and direct materials in fixed year dollars. This is sometimes called an improvement curve. The slope is calculated using hours or constant year dollars. Chapter 3, Cost Estimating Relationships, presents a more detailed explanation of improvement curve theory.

2.3.4 Production Rate

Many innovations have been made in cost improvement curve theory. One is the addition of a variable to the equation to capture the organization's production rate. The production rate is defined as the number of items produced over a given time period. This equation modifies the basic cost improvement formula to capture changes in the production rate (Q^r) and organizational cost improvement (X^b):

$$Y = AX^{b}Q^{r}$$

Where:

Y	=	Hours/unit (or constant dollars per unit)
A	=	First unit hours (or constant dollars per unit)
X	=	Unit number
b	=	Slope of the curve related to learning
Q	=	Production rate (quantity produced during the period)
r	=	Slope of the curve related to the production rate.

The equation is generally applicable only when there is substantial production at various rates. The production rate variable (Q^r) adjusts the first unit dollars (A) for various production rates during the life of the production effort. The equation also yields a rate-affected slope related to learning.

2.4 Significant Data Normalization Adjustments

The section describes some of the more complex adjustments analysts make to the historical cost data used in parametric analysis.

2.4.1 Adjustment for Consistent Scope

Adjustments are necessary to correct for differences in program or product scope between the historical data and the estimate being made. For example, suppose the systems engineering department compared five similar programs, and found that two included design-to-cost (DTC) requirements. To normalize the data, the DTC hours must be deleted from those two programs to create a data set with consistent program scope.

2.4.2 Adjustment for Anomalies

Historical cost data should be adjusted for anomalies (unusual events) when it is not reasonable to expect the new project estimates to contain these unusual costs. The adjustments and judgments used in preparing the historical data for analysis should be fully documented. For example, development test program data are collected from five similar programs, and it is noted that one program experienced a major test failure (e.g., qualification, ground test, flight test). A considerable amount of labor resources were required to fact-find, determine the root cause of the failure, and develop an action plan for a solution. A question often arises: should the hours for this program be included in the database or not? This is the kind of issue analysts must consider and resolve. If an adjustment is made to this data point, then the analyst must thoroughly document the actions taken to identify the anomalous hours.

There are other changes for which data can be adjusted, such as changes in technology. In certain applications, particularly if a commercial model is used, the model inputs could be adjusted to account for improved technologies (see the discussion of commercial models in Chapter 5, Complex Hardware Models, and Chapter 6, Complex Software Models). In addition, some contractors, instead of normalizing the data for technology changes, may deduct estimated savings from the bottom-line estimate. Any adjustments made by the analyst to account for a technology change in the data must be adequately documented and disclosed.

For instance, suppose electronic circuitry was originally designed with discrete components, but now the electronics are a more advanced technology. Or, a hardware enclosure which was made from aluminum is now made, due to weight constraints, of magnesium – what is the impact on production hours? Perfect historical data may not exist, but good judgment and analysis by an experienced analyst should supply reasonable results.

Suppose the analyst has collected four production lots of manufacturing hours data shown in the following table.

Lot	Total Hours	Units	Average hours per unit
Lot 1	256,000	300	853 hours/unit
Lot 2	332,000	450	738 hours/unit
Lot 3	361,760	380	952 hours/unit
Lot 4	207,000	300	690 hours/unit

Clearly, Lot 3's history should be investigated since the average hours per unit appear high. It is not acceptable, though, to merely "throw out" Lot 3 and work with the other three lots. A careful analysis should be performed on the data to determine why it exhibits this behavior.

2.4.3 Data Adjustment Analysis Example

Suppose the information in the following table represents a company's historical data, and that the planned system is similar to one built several years ago.

Parameter	Historical System	Planned System
Date of Fabrication	Jul 03-Jun 05	Jul 06-Dec 08
Production Quantity	500	750
Size - Weight	22 lb. external case 5 lb. internal chassis 8lb. electrical parts	20 lb. external case 5 lb. internal chassis 10 lb. electrical. parts
Volume	1 cu ft-roughly cubical 12.1 x 11.5 x 12.5	.75 cu ft-rec. solid 8 x 10 x 16.2

Parameter	Historical System	Planned System
Other Program Features	5% electrical Additional spare parts	5% electrical No spare parts

These data need several adjustments. In this example, the inflation factors, the difference in production quantity, the rate of production effect, and the added elements in the original program (spare parts) all require adjustment. The analyst must be careful when normalizing the data. General inflation factors are usually not appropriate for most situations; ideally, the analyst will have a good index of costs specific to the industry and will use labor cost adjustments specific to the company.

The quantity and rate adjustments must consider the effects of quantity changes on the company's vendors and the ratio of overhead and setup to the total production cost. Likewise, for rate factors each labor element will have to be examined to determine how strongly the rate affects labor costs. On the other hand, the physical parameters do not require significant adjustments.

The first order normalization of the historic data would consist of:

- Material escalation using industry or company material cost history;
- Labor escalation using company history;
- Material quantity price breaks using company history;
- Possible production rate effects on touch labor (if any) and unit overhead costs.

Because both cases are single lot batches, and are within a factor of two in quantity, only a small learning curve adjustment would be required. Given the schedule shown, a significant production rate adjustment is needed.

2.5 Government Evaluation Issues

DFARS 215-407-5, *Estimating Systems*, states that "contractors should use historical data whenever appropriate..." and that, "a contractor's estimating system should provide for the identification of source data and the estimating methods and rationale used to develop an estimate." Therefore, all data, including any adjustments made, should be thoroughly documented by a contractor so that a complete trail is available for verification purposes. Some key questions evaluators may ask during their review of data collection and analysis processes include:

- Are sufficient data available to adequately develop parametric techniques?
- Has the contractor established a methodology to obtain, on a routine basis, relevant data on completed projects?
- Are cost, technical, and program data collected in a consistent format?

- Will data be accumulated in a manner that will be consistent with the contractor's estimating practices?
- Are procedures established to identify and examine any data anomalies?
- Were the source data used as is, or did they require adjustment?
- Are adjustments made to the data points adequately documented to demonstrate that they are logical, reasonable, and defensible?

Chapter 7, Government Compliance, provides additional information on Government evaluation criteria.

2.6 Other Considerations

Several other issues should be considered when performing data collection and analysis.

2.6.1 Resources

Data collection and analysis activities require that companies establish sufficient resources to perform them, as well as formal processes describing data collection and analysis. Chapter 7, Government Compliance, provides information on estimating system requirements, and discusses data collection and analysis procedures.

2.6.2 Information in the Wrong Format

While the contractor may indeed possess a great deal of data, in many cases the data are not in an appropriate format to support the parametric techniques being used. For example, commercial parametric models may have a unique classification system for cost accounts that differ from those used by a company. As a result, companies using these models would have to develop a process that compares their accounting classifications to those used by the model (also known as "mapping").

In other situations, legacy systems may generate data, to meet the needs for reporting against organizational objectives, which do not directly translate into the content or format needed for cost estimating and analysis. For example, many past and existing information systems have focused on the input side with little or no provision for making meaningful translations of output data for CER development or similar types of analysis. The growing use of ERP systems, which have a common enterprise-wide database, should improve this situation. Most large organizations are implementing ERP systems, or are reengineering their existing information systems, so that parametric estimating models can easily interface with them.

2.6.3 Differences in Definitions of Categories

Many problems occur when the analyst or the database fails to account for differences in the definitions of the WBS elements across projects. Problems also occur when the definitions of the contents of cost categories fail to correspond to the definitions of analogous categories in existing databases. For example, some analysts put engineering drawings into the data category while others put engineering drawings into the engineering category. A properly defined WBS product tree and dictionary can avoid or minimize these inconsistencies.

2.6.4 The Influence of Temporal Factors

Historical data are generated over time. This means that numerous dynamic factors will influence data being collected in certain areas. For example, the definition of the content of various cost categories being used to accumulate the historical data may change as a system evolves. Similarly, inflation changes will occur and be reflected in the cost data being collected over time. As DoD deals with a rapidly changing technical environment, both cost and non-cost data generated for a given era or class of technology can quickly become obsolete. Many analysts therefore consider a data-gathering project a success if they obtain five to ten good data points for certain types of hardware.

2.6.5 Comparability Problems

Comparability problems include, but are not limited to, changes in a company's department numbers, accounting systems, and disclosure statements. They also include changing personnel from indirect to direct charge for a given function. When developing a database, the analyst must normalize it to ensure the data are comparable. For example, when building a cost database, the analyst must remove the effects of inflation so that all costs are displayed in constant dollars.

The analyst must also normalize data for consistency in content. Normalization for content ensures that a particular cost category has the same definition in terms of content for all observations in the database. Normalizing cost data is a challenging problem, but it must be resolved if a good database is to be constructed.

2.6.6 Database Requirements

Resolving database problems to meet user needs is not easy. For example, cost analysis methodologies may vary considerably from one analysis or estimate to another, and the data and information requirements for CERs may not be constant over time. An analyst’s data needs now do not determine all future needs, and must be periodically reviewed.

The routine maintenance and associated expense of updating the database must also be considered. An outdated database is of little use in forecasting future acquisition costs. The more an organization develops and relies on parametric estimating methods, the more it needs to invest in data collection and analysis activities. The contractor must balance this investment against the efficiency gains it plans to achieve through use of parametric estimating techniques. If the contractor moves towards an ERP system, the incremental cost to add a parametric estimating capability may not be significant.

Good data underpins the quality of any estimating system or method. As the acquisition community moves toward estimating methods that increase their reliance on contractor's historical costs, the quality of the data cannot be taken for granted. Industry and their Government customers should find methods to establish credible databases that are relevant to the history of the contractor. From this, the contractor will be in a better position to reliably predict future costs, and the Government to evaluate proposals based on parametric techniques.

CHAPTER 3

Cost Estimating Relationships

This chapter discusses the development and application of basic cost estimating relationships (CERs). This topic could be treated in an entire graduate-level textbook. Doing so is beyond the scope of this handbook. Although the discussion in this chapter is more in-depth than what was discussed in Chapter 1, the higher-order mathematics of CER development are relegated to Appendix B. The topic of CERs can range from the very simplistic to very complex. This chapter attempts to strike a balance. The reader needs to decide for him/herself the level of detail they will need to perform their parametric estimating assignments.

Many organizations implement CERs to streamline the costs and cycle times associated with proposal preparation, evaluation, and negotiation. The proper development and application of CERs depends on understanding the associated mathematical and statistical techniques. This chapter explains the basic and more commonly used techniques, and provides general guidance for use in developing and employing valid CERs. The discussion in this chapter:

- Identifies the differences between simple and complex CERs;
- Provides guidance on CER development, implementation, maintenance, and evaluation;
- Describes techniques for developing and implementing CERs, including linear regression ordinary least squares (OLS) "best-fit" models;
- Provides a framework for analyzing the quality or validity of a statistical model;
- Recommends procedures for developing a broad-based CER estimating capability.

The chapter also provides "rule-of-thumb" guidelines for determining the merit of statistical regression models, instructions for comparing models, and examples of simple and complex CERs.

Corporations, other types of economic enterprises, and Government cost estimating organizations make extensive use of CERs and parametric estimating models. This chapter focuses primarily on their use by Industry, as opposed to Government organizations. However, the bulk of the principles, guidelines, methods and procedures presented apply to Government cost estimating as well as to cost estimating by Industry.

3.1 CER Development

A CER is a mathematical expression, which describes how the values of, or changes in, a "dependent" variable are partially determined, or "driven," by the values of, or changes in, one or more "independent" variables. The CER defines the relationship between the dependent and independent variables, and describes how it behaves. Since a parametric estimating method relies on the value of one or more input variables, or parameters, to estimate the value of another variable, a CER is actually a type of parametric estimating technique.

Figure 3.1 demonstrates this equivalence and points out that the estimating relationship may range from simple to complex (e.g., from a ratio to a set of inter-related, multi-variable mathematical equations commonly referred to as a parametric model).

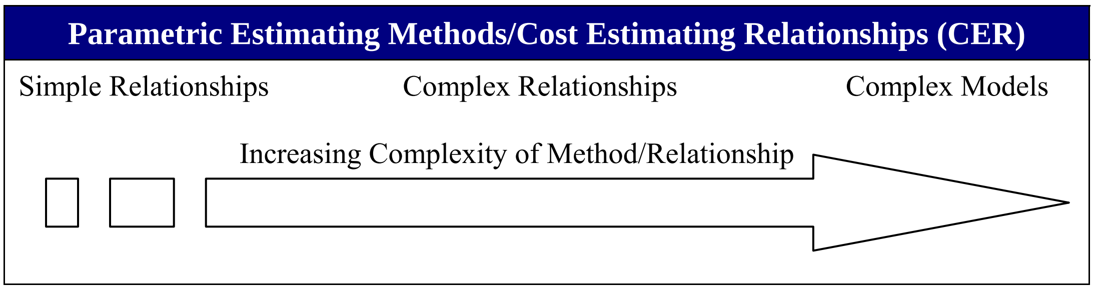


Figure 3.1 CER and Parametric Estimating Techniques

3.1.1 Cost CERs

A cost CER is one in which cost is the dependent variable. In a cost-to-cost CER the independent variables are also costs – examples are CERs which use manufacturing cost to estimate quality assurance cost, or to estimate the cost of expendable material such as rivets, primer, or sealant. The cost of one element is used to estimate, or predict, that of another.

In a non cost-to-cost relationship, the CER uses a characteristic of an item to predict its cost. Examples are CERs that estimate an item's manufacturing costs based on its weight (independent variable), or the design engineering costs from the number of engineering drawings (independent variable) involved.

It is important to note that the term "cost driver" is meant in a fairly broad sense, to include cases like those above where the "independent" variable does not actually cause the "dependent" variable to be what it is. But the two variables may be sufficiently correlated with (or "track") each other such that if one is known or estimated, then the other can be known or estimated fairly well. Thus, in the cost-to-cost CER example above, the size, quantity and complexity of the item being produced may be the real cost drivers of both the manufacturing costs <u>and</u> the quality assurance costs. The design engineering CER example illustrates true cause-and-effect behavior, where the design-engineering costs are <u>caused</u> to be what they are by the number of drawings required.

The manufacturing cost CER example is a little murkier. The item's weight and cost may correlate well, but the weight is not exactly the cause for the cost to be what it is. It is usually the basic requirements that the item must satisfy which drive both cost and weight (or size). In fact, if the requirements dictate that the item's weight be limited to the extent that unusually expensive production methods must be used, then weight per se and cost may have an inverse (i.e., negatively correlated) relationship.

Regardless of the underlying cause and effect relationships, in the context of this chapter, CER cost drivers are assumed to be either true drivers of cost or surrogates for the true cost driving requirements and constraints on the item being estimated. In many cases weight may be viewed as a good representative for most of the requirements that drive cost. In other cases it may represent cost driving requirements poorly – particularly in cases where smallness or lightness are at a premium. The same might be true for other variables that represent size or magnitude of the cost element being estimated, such as software source lines of code or processing throughput.

CERs are often used to predict labor hours, as opposed to costs. In fact, some CERs deal with normalized dependent variables, as opposed to cost or hours. For example, a CER might predict a factor, or percentage, that, when multiplied times a "base" cost, yields the cost for another work element. This approach is typically used to estimate system engineering, program management and integration, and test costs. Another example of a normalized dependent variable is the production cost/weight ratio for a type, or class, of hardware components. The ensuing discussion in this chapter applies to all of these types of CERs – whether they predict costs, labor hours,or cost estimating factors.

A cost CER is a valuable estimating tool and can be used at any time in the estimating process. For example, CERs may be used in the program concept or validation phase to estimate costs when there is insufficient system definition for more detailed approaches, such as the classical "grass roots" or "bottoms-up" methods. CERs can also be used in a later phase of a program as primary estimates or as crosschecks of non-parametric estimates. CERs may also form the primary basis of estimate (BOE) for proposals submitted to the Government or higher-tier contractors. They are also used extensively by Government agencies to develop independent cost estimates for major elements of future programs. Before developing complex parametric models, analysts typically create simple CERs which demonstrate the utility and validity of the basic parametric modeling approach to company and Government representatives.

3.1.2 Overall CER Development Process

Figure 3.2 illustrates the CER development process. The first step is the identification of an opportunity to improve the estimating process through the creation of a CER. An internal proposal or memorandum is usually prepared to describe the opportunity, data requirements, development tools, criteria for validating/accepting the CER, and the plan for using the CER and keeping it current. An organization should investigate a number of possible estimating approaches and relationships at the same time, since it is more efficient to collect data for and evaluate them as a group.

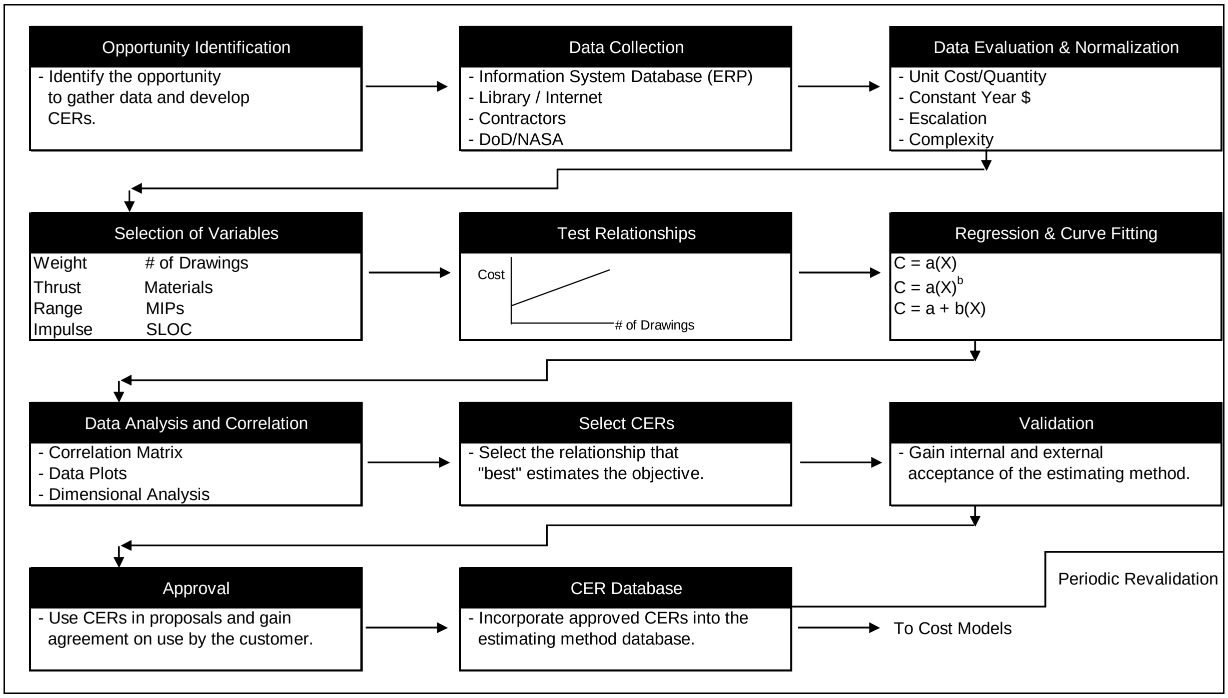


Figure 3.2 CER Development Process

3.1.3 Development Database

The value of a CER depends on the soundness of the database from which it is developed. Determination of the "goodness" of a particular CER and its applicability to the system being estimated requires a thorough analysis and knowledge of both the system and the historical data collected from similar systems. Regardless of the CER's intended application or degree of complexity, its development requires a rigorous effort to assemble and refine the data that constitutes its empirical basis. Assembling a credible database is important and, often, the most time-consuming activity in Figure 3.2. The number of valid CERs is restricted more by the lack of appropriate data than any other factor.

When developing a CER, the analyst often hypothesizes potentially useful logical estimating relationships between dependent and independent variables, and then organizes the database to test them. Another approach is where the data are collected and even organized before any relationships are hypothesized. In fact, it may be patterns in the data that suggest the most useful types of estimating relationships.

Sometimes, when assembling a database, the analyst discovers that the raw data are in the wrong format, the data displays irregularities and inconsistencies, or will not provide a good test of the hypothesis. Adjustments to the raw data, therefore, almost always need to be made to ensure a reasonably consistent, comparable, and useful set of data. Making such adjustments is often referred to as “normalizing” the raw data.

No degree of sophistication in the use of advanced mathematical statistics can compensate for a seriously deficient database. Chapter 2, Data Collection and Analysis, provides further information on collecting, organizing and normalizing CER data.

3.1.4 Testing the Logic of a CER

The developmental steps of creating a good database and hypothesizing the general form of the CER relationship are complementary. Some analysts believe the hypothesis comes first, and that this determines how and what data are collected. Others believe the reverse is true. In either case, the analyst must propose and test a logical estimating relationship, or hypothesis. For example, does it make sense to expect that costs will increase as aircraft engine thrust requirements increase? Given that it does make sense, the analyst needs to refine that hypothesis to determine whether the relationship is linear or nonlinear (curvilinear).

Framing a hypothesis involves such tasks as defining the estimating objectives for a CER, interviewing engineers to identify cost driving variables, reviewing previous technical and cost proposals, and characterizing the relationships between cost and the identified cost drivers. Only with an understanding of all the related estimating requirements should an analyst attempt to hypothesize a CER.

Note that CERs do not necessarily require robust statistical testing, although this can certainly help in most cases. Many firms use CERs and validate them by evaluating how well they predicted the final cost of that portion of the project they were designed to estimate. If the CER maintains some reasonable level of consistency, the firm continues to use it.

This “bootstrap” or “provisional” approach to testing CER validity has obvious merit, but it requires time to evolve. Statistical testing of CERs is strongly recommended prior to their use, even if the “trial by fire” approach is favored by corporate management. Regardless of the validation method, application of the technique must adhere to the company's estimating system policies and procedures. Chapter 7, Government Compliance, provides practical guidance on the Government review and evaluation criteria process.

3.2 CER Development Examples

Figure 3.3 provides examples of simple CERs implemented by various companies involved in the Parametric Estimating Initiative Reinvention Laboratories.

CER Title	Pool Description	Base Description	Application
Panstock Material	Allocated panstock dollars charged.	Manufacturing assembly “touch” direct labor hours charged.	Panstock is piece-part materials consumed in the manufacturing assembly organization. The panstock CER is applied to 100% of estimated direct labor hours for manufacturing assembly effort.
F/A-18 Software Design Support	Allocated effort required performing software tool development and support for computer and software engineering.	Computer and software engineering direct labor hours charged.	F/A-18 computer and software engineering support direct labor hours estimated for tool development.
Design Hours	Design engineering including analysis and drafting direct labor hours charged.	Number of design drawings associated with the pool direct labor hours.	The design hours per drawing CER is applied to the engineering tree (an estimate of the drawings required for the proposed work).
Systems Engineering	Systems engineering (including requirements analysis and specification development), direct labor hours charged.	Design engineering direct labor hours charged.	The system engineering CER is applied to the estimated design engineering direct labor hours.
Tooling Material	Nonrecurring, in-house, tooling raw material dollar costs charged.	Tooling nonrecurring direct labor hours charged.	The tooling material CER is applied to the estimated nonrecurring tooling direct labor hours.
Test/Equipment Material (dollars for avionics)	Material dollars (<$10k)	Total avionics engineering procurement support group direct labor hours charged.	The test/equipment material dollars CER is applied to the estimated avionics engineering procurement support group direct labor hours.

Figure 3.3 Examples of Simple CERs

3.2.1 Developing Simple CERs

For CERs to be valid, they must be developed and tested using the principles and process just discussed. Analysts rely on many forms of CERs when developing estimates, and employ them throughout the phases of the acquisition cycle. The value of a CER depends on the soundness of the database from which it was developed, and the appropriateness of its application to the estimating task. Establishing the “goodness” of a CER, and its applicability, require a thorough understanding by the cost analyst of the CER's logic and the product being estimated. A CER can take numerous forms, ranging from an informal "rule-of-thumb" or simple analogy, to a mathematical function derived from statistical analysis.

3.2.1.1 Data Collection/Analysis

When developing a CER, the analyst first concentrates on assembling and refining the data that constitute its empirical basis. A considerable amount of time is devoted to collecting and normalizing the data to ensure its consistency and comparability. More effort is usually devoted to assembling a quality database than any other task in the development process. Chapter 2 also discusses data collection and analysis. Data normalization addresses:

- **Type of effort**. This includes non-recurring versus recurring, development versus change proposals, and weapon systems versus ground support equipment.
- **Inflation**. This includes the conversion of the cost for each data point to a common year of economics or "year dollars" using established yearly company inflation rates
- **Time period covered by costs**. This includes the number of months/years in the period of performance and total cumulative data from inception to completion.
- **Measurable milestones to collect data.** This includes events such as first flight, drawing release, program completion, and system compliance test completion.

3.2.1.2 Validation Requirements

A CER, as any other parametric estimating tool, must produce, to a given level of confidence, results within an acceptable range of accuracy. It must also demonstrate estimating reliability over a range of data points or test cases. The validation process ensures that a CER meets these requirements. Since a CER developer and customer must, at some point, agree on the validation criteria for a new CER, the Parametric Estimating Reinvention Laboratory determined that the use of an integrated product team (IPT) is a best practice for reviewing and implementing it. The contractor, buying activity, DCMA, and DCAA should be part of the IPT.

Figure 3.4 illustrates the validation process flow, which incorporates the CER testing methodology discussed earlier in the chapter. The process, described in Figure 3.5, is a formal procedure which a company should use when developing and implementing a CER. It describes the activities and criteria for validating simple CERs, complex CERs, and parametric models. Figure 3.6 contains the guidelines for statistical validation (and implements the CER quality review matrix in Figure 3.4). Figure 3.7 is an example of the membership of a CER IPT designated the Joint Estimating Relationship Oversight Panel (JEROP), which is responsible for managing the processes associated with implementing, maintaining, and documenting CERs for a particular contractor.

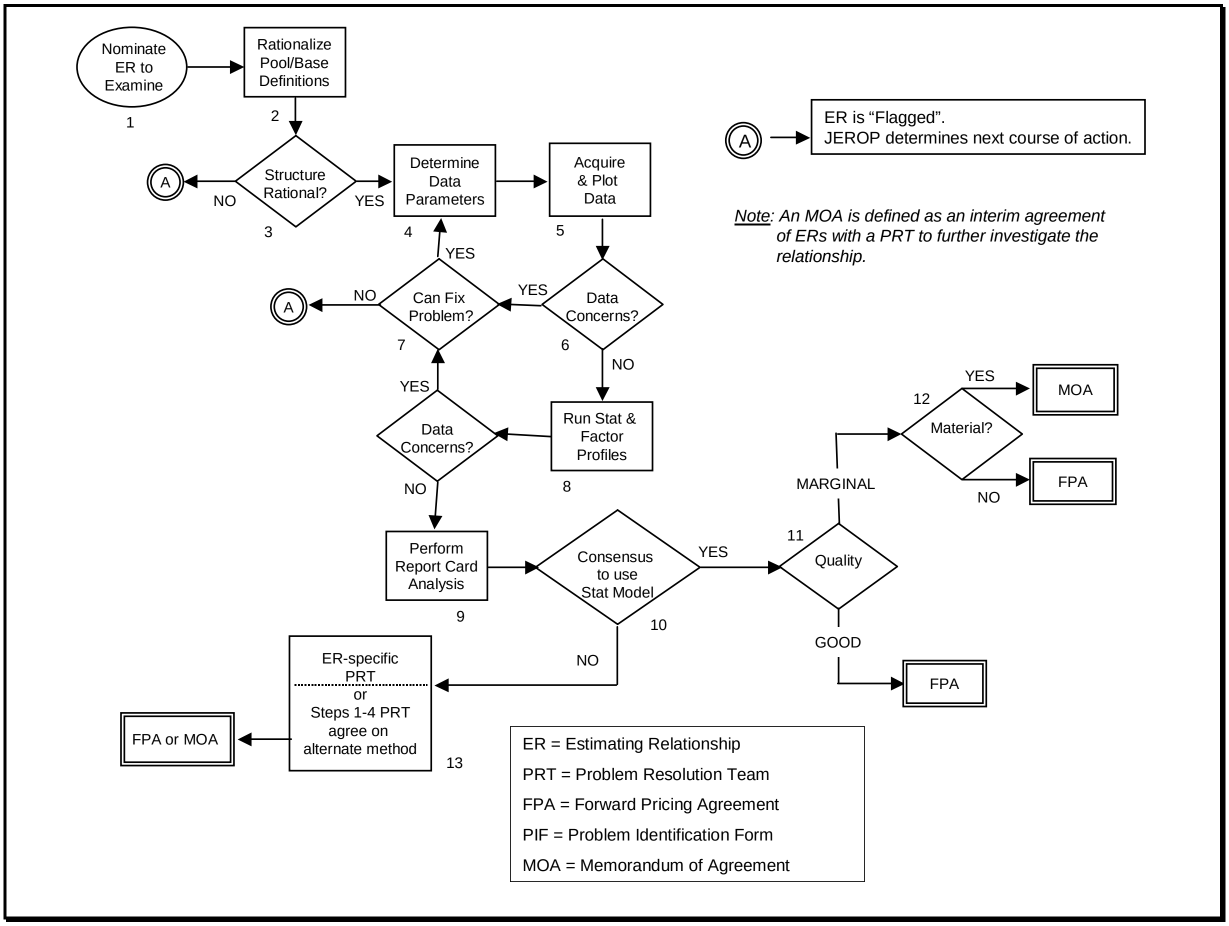


Figure 3.4 Estimating Relationship Validation Process

Discussion of Activities

1 - Assess materiality.
2 - Examine rationale and data, or use additional historical data.
3 - Investigate alternative forms.
4 - Team is encouraged to review data beyond that used to develop the current CER, i.e., additional completed jobs for Steps 1 & 2 CERs, or longer time periods for Steps 3 & 4 CERs.
  - Multivariate (more than one independent variable) solutions may be considered.
  - Examine ER across programs for rationalization of differences.
5 - Team may explore linear, logarithmic, exponential, polynomial, power, moving average, or any other model structures implied by the data patterns and/or rationale of the effort.
6 - Check for evidence of outliers, influentials, time trends, bi-modal data, etc.
7 - Team analyzes data sources.
8 - Develop results based on weighted factor methodology and linear regression with intercept, unless otherwise agreed.
  - Construct 'report card' with F-stat, R-squared, CV, and narrative for stat method; with MAD and narrative for factor method.
  - Plot results. Analyze residuals, checking for patterns in the residuals to ensure that the regression assumptions were not violated. Examine raw versus fitted data for outliers, using a rule of thumb of 2 to 3 standard deviations as a means of flagging data points for further investigation.
9 - Team analyzes 'report card' for ER based upon guidance shown in Figure 3-9.
10 - Team decides by consensus whether one or more of the methods presented are acceptable. Unless a compelling argument is presented by one of the organizations, the statistical model is to be preferred. Lack of consensus among the three organizations, or consensus that no available model is satisfactory, results in process flow to Step 13.
11 - Qualitative decision by team determining whether stat model is "Good" or "Marginal", using report card criteria as guide.
12 - Team determines materiality of the ER based on dollar impact, breadth of application, etc.
13 - Alternative methods include, but are not limited to, other statistical models, simple or weighted averaging and other factors, discreet estimating, accounting changes, investigation of other options for 'base', etc.

(A) Flagging:

A. PIF should be completed and forwarded to JEROP.
B. JEROP will determine next action, including possibility of establishing PRT to study ER.
C. If tasked with remedial investigation, PRT may:
   1. Assess materiality.
   2. Re-examine rationale and data, or use additional historical data.
   3. Investigate alternative forms.
D. PRT will determine requirements for accepting, rejecting, or restructuring ER.

Figure 3.5 Estimating Relationship Validation Process Activities

Summary of ER Report Card Criteria

This 'report card' is a summary of the key attributes of the statistically derived model and of the weighted factor, and serves as a starting point for the qualitative analysis of the proposed estimating tool.

The *p*-values of the F-test and of the t-test are the most critical, being viewed as essentially pass/fail. The other criteria, including the comments in the narrative portion of the report card, should be weighed in composite to determine the acceptability of the tool. This overall qualitative opinion should weigh the quality of the statistical results against the materiality of the effort and the quality of possible alternative methods.

		Good	Marginal
Statistically Derived ER:	*p*-value of the F-test:	≤ 0.10	≤ 0.15
	p-value of the t-test:	≤ 0.10	≤ 0.15
	Coefficient of Variation (CV) :	≤ 0.25	0.25 → 0.30
	R-squared:	≥ 0.70	0.35 → 0.70
	Narrative: *This section of the report card should be used to record other pertinent information, particularly non-quantitative information, about the effort to be modeled or about the proposed estimating tool. For example, data constraints, materiality, exogenous influences, etc., may impact the acceptability of the proposed tool.*		
Weighted Factor:	MAD as % of ER mean:	≤ 0.25	0.25 → 0.30
	Narrative: *- same as above for statistically derived model -*		

Terminology:

- F-test: Tests for trend in the data versus random dispersion.
- t-test: Measures the significance of the individual components of the model; where there is only one independent variable (one 'base' variable), the significances of the t-test and of the F-test are identical.
- R-squared: Measures the percentage of variation in the pool explained by the CER or model; varies between 0% and 100%.
- CV: Coefficient of variation is a measure of dispersion; produces a measure of 'average estimating error'.
- MAD: Mean absolute deviation is a measure of dispersion comparing how well the individual point relationships match the mean relationship of the composite data.

Figure 3.6 Summary of Estimating Relationship Report Card

JEROP Membership
Developer (Company Personnel) • Group Manager-Estimating/Systems Engineering • Principal Specialist-Estimating/Systems Engineering • Manager-Contracts & Pricing-Spares • Sr. Specialist-Accounting **DCAA** • Supervisory Auditor **DCMA** • Industrial Engineer • Contract Price Analysts The customer is not a full-time member of the IPT, but regularly provides feedback.

Figure 3.7 Joint Estimating Relationship Oversight Panel Membership

It is important to note that the IPT uses the Figure 3.6 report card as the starting point for evaluating a candidate CER. The IPT does not use the statistical tests as its only criteria for accepting or rejecting the CER. Equally important to their assessment is non-quantitative information, such as the importance of the effort or product to be estimated and the quality of possible alternative estimating methods. While statistical analysis is useful, it is not the sole basis for validating a CER, with importance also given to whether the data relationship is logical, the data used in deriving it are credible, and adequate policies and procedures for its use are in place.

3.2.1.3 Documentation

A company should document a CER to provide a clear understanding of how to apply and maintain it. The documentation, based on a standard company format, is built during the development process and includes, at a minimum, all the information necessary for a third party to recreate, validate, and implement the CER. The documentation should include:

- A clear explanation of the types of efforts or products to be estimated by the CER including:
  - Identification, explanation, and rationale for the CER database and CER functional relationships;
  - Calculation and description of effort (hours, dollars, etc.) in the pool and base.
- Information on when and how to use the CER.

- Complete actual cost information for all accounting data used. This provides an audit trail that is necessary to identify the data used.
- Noncost information (technical data).

3.2.2 Lessons Learned from CER Implementation

Simple CERs are, by their nature, straightforward in their logic and application. Figure 3.8 summarizes the lessons learned from IPTs that have implemented these CERs. Perhaps one of the most important accomplishments of the Parametric Estimating Reinvention Laboratory was the IPT partnership established between the contractor, customer, DCMA, and DCAA at each of the laboratory sites.

Cultural Change	It is important to work together openly in an IPT environment. Build trust to encourage a collaborative environment with common goals.
Empowering the IPTs	Team members should be empowered to make decisions; therefore, the teams should include people with decision-making authority.
Joint Training	All team members should participate together in training sessions. Joint IPT training provides a common understanding of terminology and techniques, and facilitates team building.
Strong Moderating	Teams should meet at regularly scheduled times and focus on the most significant issues. This may require using a trained facilitator with strong moderating skills.
Management Support	Without total commitment from management, IPTs may question the value of their efforts. Management should provide support in terms of resources, consultation, interest in the progress, resolution of stalemates, and feedback through formal communication channels.

Figure 3.8 CER IPT Lessons Learned

3.3 Curve Fitting and OLS Regression Analysis

There are two basic methods of curve fitting. In the graphical method, the analyst plots the CER data and fits a smooth curve that appears to "best-fit" the pattern of the independent and dependent variables. Although in many cases the “curve” may actually be a straight line, the vocabulary of cost estimating and mathematics describes this activity as curve fitting.

The OLS (simple linear regression) method uses mathematical formulas to develop a "best-fit" curve, and is applied using mathematical/statistical software. Although not all cost estimating relationships will be a straight line, this method is sufficiently accurate to use for many CERs with curved relationships, as well as precisely formulating the equation for a truly linear CER.

3.3.1 Graphical Method

To apply the graphical method, pairs of independent variables X, and their matching dependent variables Y, in the CER database are first plotted in the form of an X-Y "scatter diagram". Next, the analyst draws a curve (or straight line) representing the assumed CER X-Y relationship such that it passes through the approximate "middle" of the plotted data points. No attempt should be made to make the smooth curve actually pass directly through any of the data points that have been plotted. Instead, the curve should pass between the data points leaving approximately an equal number on either side of the line. The objective is to "best-fit" the curve to the data points plotted; every data point plotted should be considered equally important. The curve which is drawn then represents the CER.

Before developing a forecasting rule or mathematical equation, the analyst should plot the data in a scatter diagram. Although considered outdated for purposes of best-fitting, scatter plotting the data is still important, since it quickly gives a general idea of the relationship between the CER equation and the pattern of the data points (if any). Also, the analyst can easily focus on those data points that may require further investigation because they seem inconsistent with the bulk of the data point set. The task is easily performed with any spreadsheet or statistical software package.

3.3.2 OLS Regression Analysis

The use of one variable to predict the values of another is an application of statistical inference methods. The statistical population in this case consists of all relevant pairs of observations of the independent and dependent variables which may exist, or could be made. Generally, though, estimates or predictions are made from only a sample of that population. This is the case with CER development, where the analyst cannot collect, or even find, all the possible data (observations) which might be relevant, but still must derive a relationship from the sampling of available data.

If the relationship between the independent and dependent variables is assumed to be linear, it can be expressed by what is termed the simple regression equation:

$$Y = A + BX$$

Where:

Y represents the dependent variable (calculated from X)

X represents the independent variable

B is the slope of the line (the change in Y divided by the associated change in X), and

A is the point at which the line intersects the vertical (Y) axis (X=0).

A and B are called the parameters of the population regression line. The line, when A and B determined, represents the desired CER, since the purpose of regression analysis, and the regression line, is to provide estimates of values of the dependent variable from values of the independent variable.

Since it is usually not practical, or even possible, to obtain data for an entire population, and calculate A and B directly, the analyst must instead work with a sample collected from the population. When based on this sample, the regression line becomes Y = a + bX, where a and b are estimates of the true population parameters A and B. Since a and b are usually based on a data sample of a limited size, there always involves a certain amount of error in estimating the true values of A and B. A different sample would give different estimates of A and B.

OLS is a method for calculating the straight line (regression line) through the data set which minimizes the error involved in estimating A and B by the a and b associated with that line. OLS also provides a measure of the remaining error, or dispersion of the dependent variable values above and below the regression line, and how it affects estimates made with the regression line. Thus, the regression line which minimizes the error in estimating A and B, defined by the parameters a and b, becomes the CER.

In particular, OLS finds the best fit of the regression line to the sample data by minimizing the sum of the squared deviations of (differences between) the observed and calculated values of Y.

The observed value, Yi, represents the value that is actually recorded in the database for a given X value (Xi), while the calculated value, Yc, is the value the sample regression equation gives for the same value of X.

For example, suppose we estimated engineering hours based on the number of required drawings using the linear equation (obtained by regression analysis): EngrHours = 467 + 3.65 (NumEngrDrawings). In this case "EngrHours" is the dependent variable, and "NumEngrDrawings" is the independent variable. Suppose the company's database contained 525 hours for a program containing 15 engineering drawings. The 525 hours represents the observed value for Y when X is equal to 15. The equation however would have predicted Yc = 467 + 3.65(x) = 467 + 3.65(15) = 521.75 hours. The difference between the observed and calculated values, 3.25 hours, represents the error "e" of the OLS regression line for this data set and the data point X = 15, Y= 525.

To further define how OLS works, assume the regression line is being fit for the four points in Figure 3.9, and that the error terms, e, for these points are: (Y_1 - $Y_{C1,}$), (Y_2 - Y_{C2}), (Y_3 - Y_{C3}), (Y_4 - Y_{C4}). The line that best fits the data is the one which minimizes the sum of the squared errors, SSE:

$$\text{SSE} = \sum_{i=1}^{4} e_i^2 = (Y_1 - Y_{C1})^2 + (Y_2 - Y_{C2})^2 + (Y_3 - Y_{C3})^2 + (Y_4 - Y_{C4})^2$$

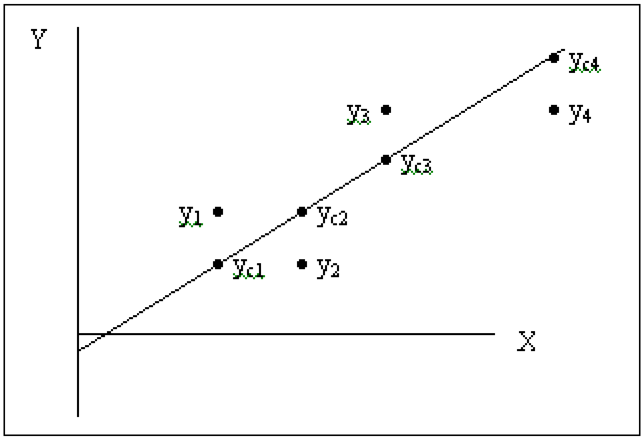


Figure 3.9 OLS Regression Analysis

The equation which minimizes the sum SSE is then the candidate regression line CER. Calculus is used to solve this classical minimization problem, yielding simple linear equations for determining the values of a and b that minimize SSE.

A good general purpose statistical software application will automatically calculate the a and b regression coefficients (CER model parameters), and provide goodness of fit statistics identified and described in Section 3.3.3 below. It will also make a scatter plot, graphing the regression equation against the CER data points.

Often, independent variables in regression analysis are also referred to as "explanatory" variables. They explain some of the variation in the Y variable via the regression equation, thereby reducing the uncertainty in estimating Y (as compared to using a simple average of all the Y data points to estimate Y). Similarly, a good regression-derived CER is said to have a high degree of "explanatory power" if it reduces the sum of the squared errors to a large degree and "explains" how the dependent variable varies as the independent variable is changed.

3.3.3 Assumptions, Limitations, and Caveats of OLS

Some assumptions, limitations, and caveats are important when interpreting and using the results of OLS regressions.

3.3.3.1 Assumptions

The mathematics of ordinary OLS regression is based on several assumptions about the underlying probability distributions of the dependent variable and probabilistic independence of the observations in the CER data set. These are not stated here but they can be found in many of the references listed in Appendix D.

Theoretically, if any of the assumptions are invalid, then the regression and CER are "flawed". Applied mathematicians, however, tend to consider the assumptions as guidelines rather than absolute rules. In most parametric cost analysis applications, the size of the CER sample is often too small to even make a conclusion about most of these assumptions. When an OLS assumption is apparently violated, the question is: "How significant is the violation?" If minor, the CER is still generally accepted as satisfactory for estimating. The size of the sum of the squared errors, SSE, and other related statistical measures described below, should provide sufficient indication of the validity of the CER even when the sample data points do not completely adhere, or appear to adhere, to the assumptions above.

3.3.3.2 Extrapolation Beyond The Range of The Observed Data

A regression equation is considered by some to be valid only over the range of data from which the sample was taken. The shape of the curve outside this range is less certain and there is more estimating risk involved. This does not mean that extrapolation beyond the range is always invalid or even a bad idea. Sometimes it is the only suitable choice available. Extrapolation beyond the range of the observed data becomes a function of the confidence the analyst has in that data. High data correlation and the expectation that the future will continue to reflect past experience makes extrapolation a reasonable approach.

The analyst must keep in mind that extrapolation assigns values (estimated costs) using a relationship for circumstances that may differ from those in the CER sample set. In any event, the larger the distance the X value of the estimate is from the center (mean) of the sampled values of X, the larger the uncertainty there is in the predicted value of Y. It is the analyst's job to decide to extrapolate using a given CER, in coordination with the technical and programmatic personnel from both the company and the Government. This can be done with the aid of OLS regression theory. The OLS "standard error of the (conditional) mean" establishes the level of uncertainty in predicted a Y value, for a given value of X, using statistics that are derived from the regression equation and the CER sample data points. The procedures for calculating standard error of the mean may be found in other references.

3.3.3.3 Cause and Effect

Regression analysis as such does not establish or identify cause and effect relationships among the regression variables. The analyst may establish cause/effect relations when testing the CER's logic, setting up the initial development hypothesis, and analyzing the database. For example, given a significant correlation between the number of telephones in a city and city liquor sales, it would be rash to assert that owning a telephone leads to buying liquor. If any cause and effect is at work in this case, it involves another variable (e.g., the size of the city's population), which drives both the number of telephones and liquor sales together. An analyst can say that cause and effect is at work in a regression CER by choosing, collecting, and testing an appropriately related data set. Also, it takes more than a good data set to establish cause and effect. The nature of the relationship, as well as good data, needs to be analyzed.

3.3.3.4 Using Past Trends To Estimate Future Trends

When using a CER, the analyst needs to identify any conditions significantly affecting the cost of products and services that are changing over time. If the original sample data are no longer relevant, due to changes in technology for example, then the CER should not be used, or must be updated based on new data.

3.3.4 Multiple Regression

In the simple regression analysis described above, a single independent variable (X) is used to estimate the dependent variable (Y), and the relationship is assumed to be linear. Multiple, or multivariate, regression considers the effect of using more than one independent variable, under the assumption that this better explains changes in the dependent variable Y. For example, the number of miles driven may largely explain automobile gasoline consumption. However, we may postulate a better explanation if we also consider such factors as the weight of the automobile.

In this case, the value of Y would be estimated by a regression equation with two explanatory variables:

$$Y_C = a + b_1X_1 + b_2X_2$$

Where:

$\mathbf{Y_C}$ is the calculated or estimated value for the dependent variable

a is the Y intercept (the value of Y when all X-variables equal 0)

$\mathbf{X_1}$ is the first independent (explanatory) variable

$\mathbf{b_1}$ is the slope of the line related to the change in X_1

$\mathbf{X_2}$ is the second independent variable

$\mathbf{b_2}$ is the slope of the line related to the change in X_2.

Finding the right combinations of explanatory variables is not easy, although the general process flow in Figure 3.2 helps. The first step involves the postulation of which variables most significantly and independently contribute toward explaining the observed cost behavior. Applied statisticians then use a technique called step-wise regression to focus on the most important cost driving variables. Step-wise regression is the process of "introducing the X variables one at a time (stepwise forward regression) or by including all the possible X variables in one multiple regression and rejecting them one at a time (stepwise backward regression). The decision to add or drop a variable is usually made on the basis of the contribution of that variable to the SSE (error sum of squares), as judged by the *F*-test."[1] Stepwise regression allows the analyst to add variables, or remove them, to determine the best equation for predicting cost.

---

[1] Gujarati, Domodar, *Basic Econometric*, New York, McGraw-Hill Book Company, 1978, p. 191.

Stepwise regression, however, requires the analyst to fully understand the variables being introduced to the model, hypothesize the effect they have, and monitor them for the effects of multicollinearity. Multicollinearity, or simply collinearity, occurs when two or more presumably independent variables exhibit a high degree of correlation with each other; that is, they are not making independent contributions towards explaining the behavior of the dependent variable.

The mathematics of regression analysis cannot easily separate or distinguish between the contributions each variable makes (when used simultaneously), which prevents the analyst from determining which variable is a better predictor (driver). The analyst must rely on the relations postulated to exist in the data, stepwise regression, and variable pair-wise correlation analysis, to determine this. In many cases, using two variables that are mildly, or possibly even heavily, correlated may be a better solution than only using one of the variables without other explanatory variables. A detailed discussion of multiple regression is beyond the scope of this handbook; Appendix D includes references to web sites and other resources which may have more information.

3.3.5 Curvilinear Regression and Cost Improvement Curve Analysis

In many cases, the relationship between the dependent variable and the independent variables may not be linear. An X-Y scatter diagram may reveal curvature in the underlying relationship when there is just one significant independent variable. However, if there is a substantial amount of variance (scatter) in the data, curvature may not be obvious. With multiple independent variables, the likelihood of "seeing" underlying curvature diminishes.

Most of the information about curvilinear regression is beyond the scope of this handbook (Appendix D lists sources which discuss it). However, parametric analysts who develop CERs should become familiar with, and able to apply, curvilinear multiple regression techniques because so many instances call for it over straight line models.

Cost improvement (learning) curve analysis, though, is a familiar cost estimating tool which uses a special form of curvilinear regression. A cost improvement curve (CIC), or "learning curve" when applied to production "hands-on" labor, represents the reduction in unit cost that typically occurs as additional units are produced. Many commercial and company hardware cost estimating models include some form of a CIC. In the following discussion, the term "learning curve" is used interchangeably with "cost improvement curve" and "CIC".

Two forms of learning curve are widely used: (1) the "unit" theory and (2) the "cumulative average" or simply "cum average", theory. Although either can be used in almost any estimating situation, the unit theory is usually used when the cost of individual production units is of interest, and the cum average is used when only the average unit "lot" cost is of interest.

The basic form of the learning curve equation is $Y = AX^b$. When a natural logarithmic transformation is applied to both sides of this equation, it is transformed to the linear form:

$$Ln(Y) = Ln(A) + b\ Ln(X)$$

Where (for both equations):

Y = Hours/unit (or constant dollars per unit)

A = First unit hours (or constant dollars

X = Unit number

b = Slope of curve related to learning

Since Ln(Y) = Ln(A) + b Ln(X) has the same form as Y = a + b(X), it can be graphed as a straight line on log-log paper, and an OLS regression analysis can be performed for it. In particular, the OLS regression equations can be used to derive the coefficients a and b from production cost data on individual units or lots. Typically several unit or lot costs are needed – say five or more.

In both cost improvement curve theories, the cost is assumed to decrease by a fixed proportion each time quantity doubles. The fixed proportion is called the "learning curve slope" or simply "learning curve", usually expressed as a percentage. For example, in the case of the unit theory, a 90 percent learning curve means that the second unit cost 90 percent of the first unit, and the fourth unit cost is 90 percent of the second unit cost, or 81 percent of the first unit cost. For the cum average theory, a 90 percent learning curve means that average unit cost of the first two units is 90 percent of the first unit cost, and the average unit cost of the first four units is 81 percent of the first unit cost.

Solving the equation Ln(Y) = Ln (A) + b Ln(X) for b, and assuming a first unit value for A = 1, and X = unit # 2, the learning curve slope is related to the learning curve coefficient by:

$$b = \frac{Ln(Slope)}{Ln(2)} \quad \text{(Note: Ln(1) = 0)}$$

Note that when, for example, the slope is 90 percent, 0.90 is used in the equation for *Slope*. The divisor Ln(2) reflects the assumption that a fixed amount of cost improvement occurs each time the production quantity doubles.

3.4 Testing the Significance of the CER

The next step in the CER development process answers these questions:

- How good – mathematically and statistically – is the CER equation?
- How useful will it be for estimating the cost of specific items or services?
- What is the confidence level of an estimate made with the CER (i.e., how likely is the estimated cost to fall within a specified range of cost outcomes)?

When the CER is based on a regression (or other statistical) analysis of the data set, the questions are best answered by reviewing the statistics of the regression line, which are a normal part of the OLS results provided by a statistics software package.

Figure 3.10 contains a list of statistics and other aspects of a candidate CER that should be evaluated whenever possible. Appendix B further defines and explains the statistics.

No single statistic either disqualifies or validates a CER. Many analysts tend to rely on two primary statistics when evaluating a CER: for example, the standard error (SE) and the adjusted coefficient of determination (R^2). Both simply measure the degree of "relatedness" between the CER's variables, but neither by itself certifies the CER as "good." However, when they are "poor" they do indicate the CER will not be an accurate predictor. All the CER statistics which are available should be studied.

Evaluation of a candidate CER begins with the data and logic of the relationship between its variables. The analyst should again ensure the accuracy of the database and verify the logic behind the CER. The data sources and accuracy should be characterized in words, as well as the logic behind the CER functional form and independent variable selections.

The analyst can then check the regression statistics, beginning with an evaluation of its variables; the t-stat for each explanatory variable indicates how important it is in the CER. One form of this statistic indicates the likelihood that the estimated variable coefficient (slope) could have resulted even though there is no underlying relationship between the variable and the dependent variable. Thus, it indicates the likelihood a "false reading" about the possibility of a relationship between X and Y.

The significance of the entire regression equation is assessed using the F-stat. Once again, it indicates the likelihood of a false reading about whether the entire regression equation exists. The F-Stat is influenced by the amount of curvature or "flatness" in the relationship between the dependent variable and the independent variables. Relationships that become relatively flat, compared to the amount of dispersion (as measured by the standard error), will have lower F-stat values than those which are steeper. Thus, the F-Stat may not be a good statistic to assess the worth of CERs, at least not on an absolute basis, when their relationships are inherently curved.

The size of the regression estimating errors is characterized by the standard error of the estimate (SE) and the coefficient of variation (CV). The SE measures the average error on an absolute basis (e.g., in units of dollars or hours). The CV is a normalized variable, typically expressing the root mean square estimating error as a percentage of the mean Y value across the CER data set points. However, neither of these statistics actually quantifies the amount of estimating error for specific values of each independent variable (this can be done with the standard error of the mean described below).

The coefficient of determination (unadjusted or adjusted, R^2) can be used to assess overall CER goodness in that it indicates the "strength" of the relationship between X and Y. However, like the F-stat, it is sensitive to the degree of CER flatness relative to the degree of dispersion.

The remaining evaluation elements in Figure 3.10 determine how good the estimates made with the CER may be. Ideally, the analyst wants a CER which has strong statistics, was developed from a large number of observations, and uses the fewest number of variables. Such a CER should produce estimates which are reasonable and have small errors.

The CER degrees of freedom (DoF) is a primary measure of CER reliability – a CER is only as reliable as the number (and quality) its data points allow. One strategy for improving CER reliability when there are relatively few data points is to merge the data set with an available data set for another similar product or service. Simple statistical methods can be used to develop a CER using the merged data set while still discriminating between costs for each of the original products/services.

The analyst should identify the actual data points for which the model poorly predicts the dependent variable. These may be viewed as "outliers" – cases that really "don't belong" to the general population. Removal of such cases should only be done with extreme care. Every reasonable effort should be made to understand why their costs are so far from the regression equation estimate. If no reasons are found to justify declaring an outlier to be a non-member of the CER population, the conventional statistical test and associated criteria might be applied as a basis for excluding the data point from the CER data set.

The analyst must also consider whether the data points used in making the estimates fall within the range of the data set; mathematically, the model is only valid over this range. In practice, use of the model is permissible outside of the range as long as the hypothesized mathematical relationship and the attendant statistical characteristics remain valid. Determining this range of validity is a judgment call, and depends on the help of engineers, analysts and others who are knowledgeable about the system being estimated. Figure 3.10 summarizes the CER statistical indicators.

t-Stat	Tests the statistical significance of the dependent variable in determining the value of the dependent variable.
F-Stat	Tests the statistical significance of the entire CER relationship.
Standard Error (SE)	Average root mean square estimating error over all the CER data points.

Coefficient of Variation (CV)	SE divided by the mean of the Y values in the CER data set. A relative, nondimensional measure of estimating error, often expressed in percentage form.
Coefficient of Determination (R^2)	Percent of the variation in the dependent variable explained by the regression relationship.
Adjusted R^2	R^2 adjusted for the number of independent variables used to explain the variation in the Y-data.
Degrees of Freedom (DOF)	Number of CER data set observations less the number of estimated parameters (number of X-variables + 1 for the constant term "a").
Outliers	Y-observations that the model predicts poorly.
Data Range	The range from the minimum X value to the maximum X value over the entire CER data set.
Standard Error of the Mean	The standard deviation of the predicted dependent variable mean value at specific values of the independent X values.

Figure 3.10 Descriptions of CER Statistical Indicators

As indicated above, the estimating error increases as the independent variables move from their mean values towards the extremes of their ranges. For this reason, it is a good idea to also calculate the standard error of the mean for estimates at the extremes of the ranges of the independent variable ranges, as well as for the mean values of each variable. This statistic depends on the number of data points from which the CER was derived and the amount of dispersion in the independent variables. It measures the estimating accuracy at particular values of the independent variables.

For CERs with the same standard error, those with data sets having more data points and robust distributions of independent variable values, over fairly broad ranges, will have less estimating error than those with fewer "bunched up", or narrow, independent variable distributions. The standard error of the mean can contribute to establishing extrapolation limits by establishing estimating uncertainty levels beyond the original ranges of the CER data. It, along with the SE, is also instrumental in establishing the uncertainty in a single estimate made with the CER and specific values of each independent variable as opposed to the mean of many estimates.

There are no definite standards prescribing pass/fail criteria for the CER and its various statistics. The validation of a given CER is based on discussions between contractor and customers, statistics, the data collection and normalization, intended application, and the CER's logic. These together form the basis for accepting or rejecting the CER. Finally, to keep perspective on the evaluation criteria, the analyst must always ask: "If I reject this CER as the basis for estimating, is the alternative method any better?"

3.5 When to Use a CER

When a CER has passed its evaluation, it is ready for application. A CER may be used as a primary estimating method to forecast costs, or to cross check an estimate developed using another estimating technique. For example, an analyst may have generated an estimate using a grassroots approach (e.g., a detailed build-up by hours and rates), and then used a CER estimate based on the same data as a sanity test of the grassroots' results. A regression CER can provide more realistic estimates than grass roots approaches if the latter are not closely and objectively tied to actual cost history.

A CER developed to make a specific forecast may be used with far more confidence than a "generic" CER developed for a wider range of applications. Care must be especially taken in using a generic CER when the characteristics of the forecasting universe are, or are likely to be, different from those of the CER database used to build it. A generic CER may have to be revalidated or modified for use in a particular application, and the changes made to it documented.

To be able to apply good judgment in the use of CERs, the analyst needs to know their strengths and weaknesses.

3.5.1 Strengths

- CERs can be excellent predictors when implemented correctly, and they can be relied upon to produce quality estimates when used appropriately.
- Use of valid CERs can reduce proposal preparation, evaluation, negotiation costs, and cycle time, particularly with regard to low-cost items that are time and cost intensive to estimate using other techniques.
- They are quick and easy to use. Given a CER equation and the required input parameters, developing an estimate is a quick and easy process.
- Most CERs can be used with a small amount of top-level information about the product or service being estimated. Consequently, CERs are especially useful in the research, development, test and evaluation (RDT&E) phase of a program.

3.5.2 Weaknesses

- CERs may be too simple to be used to estimate certain costs. When detailed information is available, a detailed estimate may be more reliable than one based on a CER.
- Problems with the database may mean that a particular CER should not be used. While the analyst developing a CER should also validate both the CER and the database, it is the responsibility of the parametrician to determine whether it is appropriate to use a CER in given circumstances

by reviewing its source documentation. The user should determine what the CER is supposed to estimate, what data were used to build it, how current are the data, and how the data were normalized. Never use a CER or cost model without reviewing the source documentation.

3.6 Examples of CERs in Use

This and the following section contain CER examples provided by contractors who participated in the Parametric Estimating Reinvention Laboratory. A CER calculates changes in prices or costs (in constant dollars) as some physical, performance, or other cost-driving parameter changes. Such a relationship may be applied to a variety of items and services.

3.6.1 Construction

Many construction contractors use a rule of thumb that relates floor space to building cost. Once a general structural design is determined, the contractor or buyer can use this relationship to estimate total building price or cost, excluding the cost of land. For example, when building a brick two-story house with a basement, a builder may use $60/square foot to estimate the price of the house. Assume the plans call for a 2,200 square foot home. The estimated build price, excluding the price of the lot, would be $60/sq. ft. x 2,200 sq. ft. = $132,000.

3.6.2 Electronics

Manufacturers of certain electronic items have discovered that the cost of a completed item varies directly with the number of total electronic parts in it. Thus, the sum of the number of integrated circuits in a specific circuit design may serve as an independent variable (cost driver) in a CER to predict the cost of the completed item. Assume a CER analysis indicates that $57.00 is required for set-up, and an additional cost of $1.10 per integrated circuit required. If evaluation of the engineering drawing revealed that an item was designed to contain 30 integrated circuits, substituting the 30 parts into the CER gives:

Estimated item cost = $57.00 + $1.10 per integrated circuit * number of integrated circuits

= $57.00 + $1.10 (30)

= $57.00 + $33.00

= $90.00

3.6.3 Weapons Procurement

CERs are often used to estimate the cost of the various parts of an aircraft, such as that of a wing of a supersonic fighter. Based on historical data, an analyst may develop a CER relating wing surface area to cost, finding that there is an estimated $40,000 of wing cost (for instance, nonrecurring engineering) not related to surface area, and another $1,000/square foot that is related to the surface area of one wing. For a wing with 200 square feet of surface area:

Estimated price = \$40,000 + (200 sq ft x \$1,000 per sq. ft.)
= \$40,000 + 200,000
= \$240,000

3.7 CERs from the Defense Procurement and Acquisition Policy

The following is an excerpt from the Defense Procurement and Acquisition Policy (DPAP), *Developing and Using Cost Estimating Relationships.* It's useful to understand what the DPAP sees as important to the process of CER development.

3.7.1 Cost Estimating Relationship Definition

As the name implies, a cost estimating relationship (CER) is a technique used to estimate a particular cost or price by using an established relationship with an independent variable. If you can identify an independent variable (driver) that demonstrates a measurable relationship with contract cost or price, you can develop a CER. That CER may be mathematically simple in nature (e.g., a simple ratio) or it may involve a complex equation.

3.7.2 Steps for Developing a Cost Estimating Relationship

Strictly speaking, a CER is not a quantitative technique. It is a framework for using appropriate quantitative techniques to quantify a relationship between an independent variable and contract cost or price.

Development is a 6-step process. Follow the six steps whenever you develop a CER. Whenever you evaluate a CER developed by someone else, determine whether the developer followed the six steps properly.

Step 1. Define the dependent variable (e.g., cost dollars, hours, and so forth.) Define what the CER will estimate. Will the CER be used to estimate price, cost dollars, labor hours, material cost, or some other measure of cost? Will the CER be used to estimate total product cost or estimate the cost of one or more components? The better the definition of the dependent variable, the easier it will be to gather comparable data for CER development.

Step 2. Select independent variables to be tested for developing estimates of the dependent variable. In selecting potential independent variables for CER development:

- Draw on personnel experience, the experience of others, and published sources of information. When developing a CER for a new state-of-the-art item, consult experts experienced with the appropriate technology and production methods.
- Consider the following factors:
    - Variables should be quantitatively measurable. Parameters such as maintainability are difficult to use in estimating because they are difficult to measure quantitatively.

  - Data availability is also important. If you cannot obtain historical data, it will be impossible to analyze and use the variable as a predictive tool. For example, an independent variable such as physical dimensions or parts count would be of little value during the conceptual phase of system development when the values of the independent variables are not known. Be especially wary of any CER based on 2 or 3 data observations.
  - If there is a choice between developing a CER based on performance or physical characteristics, performance characteristics are generally the better choice, because performance characteristics are usually known before design characteristics.

Step 3. Collect data concerning the relationship between the dependent and independent variables. Collecting data is usually the most difficult and time-consuming element of CER development. It is essential that all data be checked and double checked to ensure that all observations are relevant, comparable, relatively free of unusual costs.

Step 4. Explore the relationship between the dependent and independent variables. During this step, you must determine the strength of the relationship between the independent and dependent variables. This phase of CER development can involve a variety of analytical techniques from simple graphic analysis to complex mathematical analysis. Simple ratio analysis, moving averages, and linear regression are some of the more commonly used quantitative techniques used in analysis.

Step 5. Select the relationship that best predicts the dependent variable. After exploring a variety of relationships, you must select the one that can best be used in predicting the dependent variable. Normally, this will be the relationship that best predicts the values of the dependent variable. A high correlation (relationship) between a potential independent variable and the dependent variable often indicates that the independent variable will be a good predictive tool. However, you must assure that the value of the independent variable is available in order for you to make timely estimates. If it is not, you may need to consider other alternatives.

Step 6. Document your findings. CER documentation is essential to permit others involved in the estimating process to trace the steps involved in developing the relationship. Documentation should involve the independent variables tested, the data gathered, sources of data, time period of the data, and any adjustments made to the data.

3.7.3 Identifying Situations for Use

You can use a cost estimating relationship (CER) in any situation where you quantify one of the following:

- A relationship between one or more product characteristics and contract cost or price. A product-to-cost relationship uses product physical or performance characteristics to estimate cost or product price. The characteristic or characteristics selected for CER development are usually not the only ones driving cost, but the movement of cost has been found to be related to changes in these characteristics.

- A relationship between one or more elements of contract cost and another element of contract cost or price. A cost-to-cost relationship uses one or more elements of contract cost to estimate cost or product price. If you can establish a relationship between different elements of cost (e.g., between senior engineering labor hours and engineering technician hours), you can use a CER to reduce your estimating or analysis effort while increasing accuracy. If you can establish a relationship between an element of cost and total price (e.g., between direct labor cost and total price), you can use that information to supplement price analysis, without requiring extensive cost information.

3.7.4 Developing and Using Estimating Factors

An estimating rate or factor is a simple ratio, used to estimate cost or price. The rule of thumb used to develop table price estimates in the previous section is an example – $19 per square foot. As the size of the table top increases, the price estimate increases in direct proportion. Most rules of thumb are simple factors. Many CERs developed by Government or Industry are also simple factors. They are relatively easy to develop, easy to understand, and in many cases quite accurate.

Development and use of estimating rates and factors involves two important implicit assumptions.

- There is no element of the cost or price being estimated that is not related to the independent variable (i.e., there is no "fixed cost" that is not associated with the independent variable).
- The relationship between the independent variable and the cost being estimated is linear.

If you believe that there are substantial costs that cannot be explained by the relationship or that the relationship is not linear, you should either try to develop an equation that better tracks the true relationship or limit your use of the estimating factor to the range of the data used in developing the factor.

3.7.5 Developing and Using Estimating Equations

Not all estimating relationships lend themselves to the use of simple estimating factors. If there is a substantial element of the cost or price being estimated that is not related to the independent variable (i.e., there is a "fixed cost" that is not associated with the independent variable), you should consider using a linear estimating equation. If the relationship is not linear, consider a nonlinear estimating equation.

CERs, like most other tools of cost analysis, MUST be used with judgment. Judgment is required to evaluate the historical relationships in the light of new technology, new design, and other similar factors. Therefore, a knowledge of the factors involved in CER development is essential to proper application of the CER. Blind use of any tool can lead to disaster.

3.7.6 Identifying Issues and Concerns

As you perform price or cost analysis, consider the issues and concerns identified in this section as you consider use of a cost estimating relationship.

- Does the available information verify the existence and accuracy of the proposed relationship?

  Technical personnel can be helpful in analyzing the technical validity of the relationship. Audit personnel can be helpful in verifying the accuracy of any contractor data and analysis.

- Is there a trend in the relationship?

  For example, the cost of rework is commonly estimated as a factor of production labor. As production continues, the production effort should become more efficient and produce fewer defective units which require repair. The factor should decrease over time. You should also consider the following related questions: Is the rate distorted by one bad run? What is being done to control the rate? What else can be done?

- Is the CER used consistently?

  If an offeror uses a CER to propose an element of cost, it should be used in all similar proposals. Since the CER can be used to estimate the average value, some jobs should be expected to cost more and others less. With a valid CER, you assume the variances will be minor and will average out across all contracts. To use a CER in some cases and a discrete estimate in others destroys it usefulness by over or understating costs across all proposals (e.g., using the average unless a discrete estimate is lower/higher negates the averaging out of cost across all contracts and is clearly unfair to one of the contracting parties).

- Has the CER been consistently accurate in the past?

  No matter how extensive the price/cost information or how sophisticated the analysis technique, if a CER does not do a good job of accurately projecting cost, then it is not a useful tool.

- How current is the CER?

  Even the most accurate CER needs to be reviewed and updated. While the time interval between updates will differ with CER sensitivity to change, in general a CER should be reviewed and updated at least annually. A CER based on a moving average should be updated whenever new data become available.

- Would another independent variable be better for developing and applying a CER?

  If another independent variable would consistently provide a more accurate estimate, then it should be considered. However, remember that the CER may be applicable to other proposals, not just yours. It is possible that a relationship which works well on your contract would not work well across the entire contract population. When assessing CER validity, you should consider all affected contracts.

- Is the CER a self-fulfilling prophecy?

  A CER is intended to project future cost. If the CER simply "backs into" a rate that will spread the cost of the existing capacity across the affected contracts, then the CER is not fulfilling its principle function. If you suspect that a CER is being misused as a method of carrying existing resources, you should consider a should-cost type review on the functions represented by the CER.

- Would use of a detailed estimate or direct comparison with actuals from a prior effort produce more accurate results?

  Development of a detailed estimate can be time consuming and costly but the application of the engineering principles required is particularly valuable in estimating cost of efficient and effective contract performance.

3.8 Evaluating CERs

3.8.1 Government Evaluation Criteria

Chapter 7, Government Compliance, discusses the requirements of an estimating system and also discusses estimating system requirements and evaluation criterion in detail. Government evaluators evaluate and monitor CERs to ensure they are reliable and credible cost predictors. This section provides a general overview of CER evaluation procedures, which generally include:

- Determining if the CER relationships are logical;
- Verifying that the data used are adequate;
- Performing analytical tests to determine if strong statistical relationships exist;
- Ensuring CERs are used consistently with established policies and procedures, and that they comply with all Government procurement regulations.

3.8.2 Logical Data Relationships

When analyzing a CER, evaluators must determine that the functional relationship it assumes between the cost drivers (explanatory variables) and the cost element to be estimated is logical. Drivers which affect the cost element may be identified through a number of sources, including personal experience and published data and studies. One of the Parametric Estimating Reinvention Laboratory IPTs developed a process for determining possible cost drivers; using brainstorming techniques, it first identified potential drivers, and then surveyed experts for their opinion of the merit of each one. Figure 3.11 is an example of the survey they developed.

PARAMETRIC: TOOLING MATERIAL COSTS (IN-HOUSE TOOLING)
Cost of raw materials and parts which are purchased to fabricate tools in-house.

If you don't know - respond with " ? "

	Proposed Cost Driver	Metric Proposed	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
1	changes	Count of part design changes (start tool Fabrication to final tool buyoff)	YN	YN	YN	NN	EH	ML	XH	EM
2	design hours	Number of hours to design tool	NN	NN	NN	YY	ML	HL	MM	MM
3	experience (tool des)	Years experience of tool designer	YN	YN	YY	NN	EA	EL	HL	HL
4	number of tools	Count of total number of tools to be built	NN	YY	YY	YY	HM	HM	HM	HM
5	production run	Number of parts tool is designed to build (i.e., 500 parts to be built using tool)	YY	YY	YY	YN	ML	ML	MM	MM
6	rework	Total number of (fabrication) rework orders for a particular tool during initial build	YY	YY	YY	YY	HM	MM	MM	MM
7	schedule *	Measure of compression of flowtime to produce tool *.	YN	YN	YN	YN	EE	EE	EH	EH
8	subsystems	Aircraft Subsystem category (i.e. tool builds part "A" which is in subsystem "X")	YN	YN	YN	YN	EE	EE	EH	EH
9	complexity	Measure of tool complexity	YY	YY	YY	YN	EM	EM	EH	EH
10	speeds	Measure of speed of moving parts on a tool	NN	NN	NN	NN	EH	EH	EH	EH
11	type	Type of tool	YY	YY	YY	YY	MM	MM	ML	MM
12	weight	Weight of tool	YN	YN	YN	NN	EH	EH	EH	EH
13	material type	Type of material the tool is made of (steel, alum., graphite, fiberglass...)	YY	YY	YY	YN	EM	EM	EM	EM

GOOD
MEDIUM
BAD

Q1 Do you think that this would be a good predictor of Tool material costs? (Y/N)

Q2 Is this a determinant of Tool material cost? (direct and logical relationship) (Y/N)

Q3 Would you expect any correlation between this item and Tool material cost? (Y/N)

Q4 Is this type of information typically recorded and available? (Y/N)

Q5 What is the difficulty in obtaining HISTORICAL information? (X-impossible, E-extremely high, H-high, M-medium, L-low, A-readily avail.)

Q6 What is the COST of obtaining HISTORICAL information? (E-extreme (yrs), H-high (mo's), M-medium (wk's), L-low (days), A-almost none)

Q7 What is the difficulty in obtaining CURRENT / FUTURE information? (X-impossible, E-extremely high, H-high, M-medium, L-low, A-readily avail.)

Q8 What is the COST of obtaining CURRENT / FUTURE information? (E-extreme (yrs), H-high (mo's), M-medium (wk's), L-low (days), A-almost none)

Figure 3.11 Cost Driver Survey Example

Using this survey, the IPT identified those cost drivers which had the most effect on a given cost element, and were therefore candidates for further analysis. The IPT used these key questions, which are important to any CER evaluator:

- Does the CER seem logical (e.g., will the cost driver have a significant impact on the cost of the item being estimated)?
- Will the cost driver be a good predictor of cost?
- How accessible are the data needed to develop the CER (both cost and non-cost)?
- How much will it cost to obtain the necessary data?
- How much will it cost to obtain the data in the future?

- Will there be a sufficient number of data points to implement and test the CER(s)?
- Have all potential cost drivers been considered?
- Were any outliers excluded and, if so, why?

3.8.3 Credible Data

All data collected to support parametric estimating tools must be accurate and their sources documented. An evaluator should verify the integrity of the data, and the adjustments made during their normalization. Some questions which should be asked during an evaluation include:

- Were sufficient data available to adequately develop parametric techniques?
- Has the contractor established a methodology to obtain, on a routine basis, relevant data on completed projects?
- Are cost, technical, and programmatic data collected in a consistent format?
- Are procedures established to identify and examine data anomalies?

3.8.4 Strength of CER Relationships

After determining that the CER relationships are logical and the data used to develop the CER are credible, the evaluation next assesses the strength of the relationship between the cost and driver variables. This can be tested with a number of quantitative techniques, such as simple ratio analysis, analysis of variance, and other statistical analysis. The evaluation tools used should be based on the number of data points available for testing as well as the importance of the cost estimate. When a company uses simple factors, for example, based on prior program experience to estimate the costs of minor items or services, a simple evaluation technique (e.g., comparisons with previous estimates) is best. When sufficient data is available, and especially when the cost to be estimated is significant, some form of statistical analysis should be used.

3.8.5 CER Validation

CER validation is the process, or act, of demonstrating the technique's ability to function as a credible estimating tool. Validation includes ensuring contractors have effective policies and procedures, data used are credible, CERs are logical, and CER relationships are strong. Evaluators should test CERs to determine if they can predict costs within a reasonable degree of accuracy. The evaluators must use good judgment when establishing an acceptable range for accuracy. Generally, CERs should estimate costs as accurately as other estimating methods (e.g., bottoms-up estimates). This means when evaluating the accuracy of CERs to predict costs, assessing the accuracy of the prior estimating method is a key activity.

CER validation is an on-going process. The evaluation should determine whether contractors using CERs on a routine basis have a proper monitoring process established to ensure CERs remain reliable. A best practice is to establish ranges of acceptability, or bands, to monitor the CERs. If problems are identified during monitoring, contractors should have procedures in place to perform further analysis activities. In addition, when a contractor expects to use CERs repeatedly, the use of forward pricing rate agreements (FPRAs) should be considered. FPRAs are discussed in Chapter 7, Government Compliance.

3.8.6 Summary of CER Evaluation

The following list suggests additional questions which might be asked about a simple CER to determine its limitations and applicability. They could also be asked about a complex CER, or the group of CERs in a model, to help determine their scope and usefulness for large procurements. Consider the importance of the costs which a CER estimates when using the questions. Don't spend a lot of time asking them, or getting their answers, for example, when the CER's result is a minor cost, or is lost in rounding when rolled into higher-level estimate.

1. What proportion of the estimate is directly affected by the CER?
2. How much precision is needed for the total estimate and for the part of it affected by the CER?
3. Is there a logical relationship between a CER's dependent variable and its independent variables?
4. Is this relationship functional or statistical? If functional, what is it, and why? If statistical, does the associated data support the CER's intended application?
5. Are relationship and the independent variables statistically significant? At what level of confidence?
6. What happens to the estimate when reasonable variations of the input parameters are used?
7. Are the analytical methods and techniques used to develop and use the CER sound and appropriate?
8. Does the CER generate the type of estimate required?
9. Are the model input parameters available and reliable in the phases of the system life cycle when it will be used?
10. Are the concepts behind the CER widely accepted in Industry and generally understood?
11. Are the CER's strengths and limitations reasonable?
12. What is the effect of input uncertainty on the estimate's confidence interval?
13. Are the mathematical procedures used to develop the CER rigorous?

14. Does the CER integrate information from other systems?
15. Is the CER compatible with other CERs/models in theory and operation?
16. Is a sufficient amount of accurate and relevant historical data available for model development?
17. Are the cost estimates made with the model consistent with user/contractor performance?
18. Does the CER model documentation provide insight into historical data?
19. What parametric development concepts does the CER incorporate?
20. Are the developing organization's estimating systems and policies current?
21. Are the CER's source data verifiable?
22. Does the developing organization have written guidelines for the development and support of parametric estimates?
23. How are users trained to use the CER?
24. How is the CER updated?
25. Do the CER's parameters adequately describe the item/service which is estimated?
26. Are the engineering input decisions that contributed to the CER development documented?
27. How difficult is it to use the CER?
28. Is the CER flexible (e.g., to changing programmatic and technical issues, or parameters)?
29. Is the CER model useful at varying levels of input detail?
30. Can the CER be used across a range of time, products, and technology changes?
31. How easy is it to misuse the CER?
32. Does the CER avoid personal or organizational bias?
33. Can the CER results be adjusted?
34. Does use of the CER require experienced analysts and/or special training?
35. Have the CER's results been checked against test cases?
36. Are the CER's results in the correct format and level of detail?

CHAPTER 4

Company Developed Complex Models

This chapter provides practical information about developing, deploying, and maintaining company developed parametric models. Company developed models, also referred to as company-owned, in-house, or proprietary models, differ from cost estimating relationships (CERs) because of their higher level of complexity and the range of costs they estimate. Unlike commercial models, company developed models are designed for the specific estimating needs of an organization or to describe a particular product. This chapter focuses on the special tasks and concerns of model building, which are beyond those found in CER development, and draws upon contractor examples of company developed models. It also looks at the effort involved in implementing and maintaining these models.

The results of the Parametric Estimating Reinvention Laboratory demonstrated that the best investment a company can make when embarking on a proprietary model development effort is joint planning among internal management, external customers, and Government representatives. For that purpose, this chapter:

- Discusses issues companies should consider prior to their implementation of a model;
- Provides a process flow diagram that illustrates the model development process, and highlights key issues related to the implementation of company developed models;
- Explains the processes involved in validating a proprietary model, including examples from Parametric Estimating Reinvention Laboratory participants;
- Highlights criteria that can be used by evaluators to review company developed models.

The information in this chapter also applies to special purpose models developed by Government agencies, if those models otherwise have the same features as company developed ones.

4.1 Background

Companies develop their own parametric models for a variety of reasons. For example:

- They have specific estimating needs that cannot be achieved by using an existing commercial parametric model.
- Some firms, after experiencing success with CERs, expand their estimating tool set to include more complex parametric models, sometimes tailored to a specific purpose. For example, parametric models can be used to prepare estimates for significant portions of a proposal (e.g., sub-assemblies, program management, systems engineering) or an entire proposal, including the project's total life cycle cost. The proper use of a validated, company developed model should increase the efficiency of the estimating process as well as improve the quality, accuracy, and consistency of the resulting estimates.
- In-house models protect sensitive information, whether proprietary or classified, and may be developed using a WBS different from that of commercial models.
- Management may not be willing to "bet the company" on estimates produced by a commercial model whose CERs are not well presented nor understood.

4.1.1 General Definitions

Parametric models can generally be classified as commercial or company developed, and this chapter will refer to the latter as proprietary models. Complex parametric models may consist of many interrelated CERs, as well as other equations, ground rules, assumptions, and variables that describe and define the situation being studied.

Models generate estimates based upon certain input parameters, or cost drivers. Parameters "drive the cost" of the end product or service being estimated. Some examples are weight, size, efficiency, quantity, and time. Some models can develop estimates with only a limited set of descriptive program inputs; others, however, require the user to provide many detailed input values before the model can compute a total cost estimate. A model can utilize a mix of estimating methods, and it may allow as inputs estimates from other pricing models (or information systems) or quotes from external sources, such as subcontracts.

Commercial parametric estimating models, available in the public domain, use generic algorithms and estimating methods which are based on a database that contains a broad spectrum of industry-wide data. Because this data encompasses many different products, a company working with a commercial parametric model must calibrate it before using it as a basis of estimate (BOE) for proposals submitted to the Government or higher-tier contractors. Calibration tailors the commercial model so it reflects the products, estimating environment, and business culture of that particular company. Chapter 5, Complex Hardware Models, and Chapter 6, Complex Software Models, discuss commercial parametric models.

A proprietary model offers an alternative to trying to use a commercial model to meet an organization's unique estimating requirements. Proprietary models are developed for an organization's own product and cost estimating needs and are, in effect, self-calibrated.

4.1.2 Examples of Proprietary Models

Proprietary models can be implemented for a variety of estimating purposes, and have a wide range of complexity, completeness, and application, as these examples demonstrate.

4.1.2.1 Forward Pricing Rate Model

This model was developed by a contractor to streamline its estimating practices for calculating forward pricing rates (including overhead and general and administrative (G&A) expense rates). The model calculates rates based on five business activities (cost drivers): cost-type sales; fixed price sales; proprietary sales; bid and proposal (B&P) costs; and independent research and development (IR&D) costs. Chapter 8, Other Parametric Applications, provides additional information on this type of model.

4.1.2.2 Program Management Model

This model, called E-PROMM, was created to establish a repeatable methodology for estimating program management costs. The model relies on a relationship between program management costs and the combined cost of engineering hours, manufacturing hours, and material dollars. The model allows for a series of program descriptor parameters that adjust how the next project differs from the nominal program in the database.

4.1.2.3 Space Communications Payload Cost Model

This parametric model was designed to establish a standard methodology for estimating non-recurring (development plus one qualification model) and recurring (theoretical first unit, or T1, based on a 95% learning rate) cost for space-qualified communications payloads. The model WBS was derived from a composite of other product-based cost models. The CERs were statistically derived from historic cost data obtained from completed space programs, including ACTS, DSCS, INTELSAT, UFO, and Milstar. Typical cost drivers are subsystem weight in pounds, antenna area in square inches, and transmitter operating power in watts.

Figure 4.1 shows an overview of the model. In cases where a statistically significant CER could not be developed, an average value was provided.

Companies use this model to make sanity check estimates for major engineering proposals or high-level engineering trade studies.

WBS	Non-Recurring CER (FY2006K$)	Recurring CER T1 (FY2006K$)
Receiver	2449.5 + 431.9 * Wt	1875.9 + Wt ^ 2.42
Transmitter (SSA)	2385.6 - 75.9 * Wt	933.1 - 103.6 * Wt + 17.9 * Operating Power
Transmitter (TWTA)	5260 (avg value)	1036.2 + 81.9 * Wt
Transponder	2780 (avg value)	-453 + Wt ^ 2.25
Antenna (Reflector)	1225.6 + 0.41 * Area	573.5 + (Area/Wt) ^ 1.45
Antenna (Horn)	1334 (avg value)	-199.8 + 94.2 * Wt
Space-borne Electronics	10259 (avg value)	-1350.9 + 198 * Wt
Waveguides	1353 (avg value)	10.9 + 14.6 * Wt
Power Dividers	1353 (avg value)	192.9 + 47.4 * Wt

Figure 4.1 Example of Model with CERs Matched to Product

4.1.2.4 Space Sensor Cost Model

This complex model was also designed to establish a standard methodology for estimating nonrecurring and recurring costs for a space sensor payload, but offers greater flexibility than the Space Communications Payload model in the scope of its CERs. It provides individual estimates for the engineering and qualification units comprising the nonrecurring costs as well as individual estimates for the production setup and flight unit T1 comprising the recurring costs. The model is discussed later in this chapter.

4.1.2.5 Follow-On Production Model and Estimating Tool

This model generates the total recurring production costs for multiple individual product lines. This model has the ability to estimate range-quantity costs for multi-year or indefinite delivery indefinite quantity (IDIQ) procurements. This model has a number of modules: material; assembly; inspection and test; manufacturing support; engineering support; program schedule; rough order of magnitude cost; and proposal documentation. It has been modeled to five individual product lines. This chapter provides more information in a later section.

4.2 The Model Development Process

Figure 4.2 shows a process flow diagram that highlights the major activities involved in developing a proprietary model. This section provides detail on each of these steps, and includes examples from Parametric Estimating Reinvention Laboratory teams as well as companies that implemented proprietary models for use in developing proposal estimates.

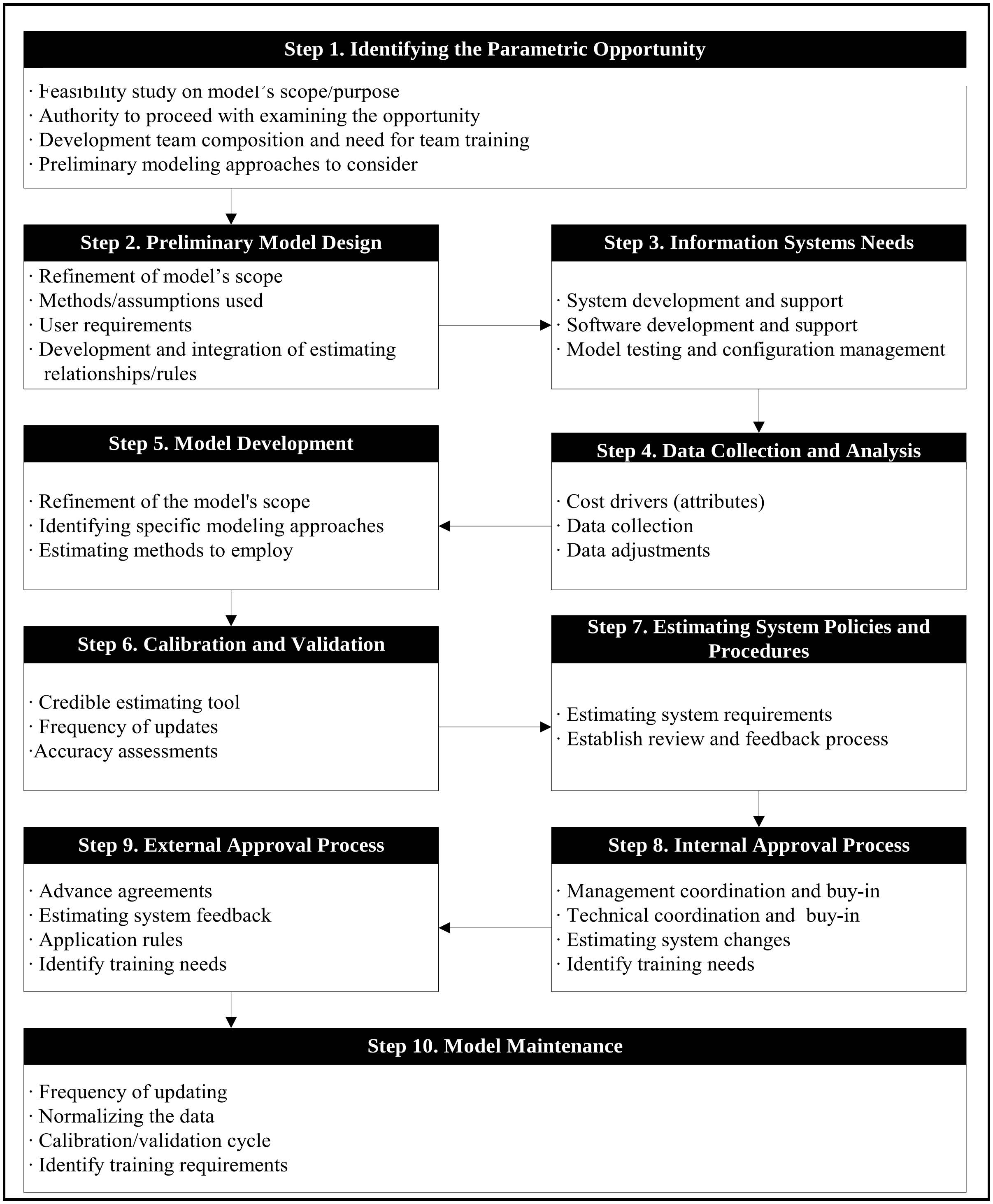


Figure 4.2 Typical Model Development Process – Company Developed Models

4.2.1 Step 1: Identifying the Parametric Opportunity

One of the most critical steps in the proprietary model development process is the identification of a good opportunity for implementing a parametric model. This involves two points. First, it is important to investigate the feasibility of developing the model, which entails an evaluation of both its technical feasibility and cost effectiveness. Technical feasibility refers to the ability of the model to meet the estimating needs of the organization, and examines whether the organization has the resources to develop the model within a reasonable timeframe. This includes performing a cost-benefit analysis to decide whether a proprietary model would be cost-effective to implement and maintain.

All potential benefits should be considered in the cost-benefit analysis; for example, contractors have achieved significant savings in proposal preparation, evaluation, and negotiation through the implementation of proprietary parametric estimating models. Other contractors have achieved additional benefits through multiple applications of the same model, such as for design studies, target costing, and contract risk management as well as basic estimating.

The second critical point involves gaining the support of internal upper-level (including program) management, key customer management, and local Government representatives, particularly from the DCMA and the DCAA. If the model then meets the acceptance criteria provided by these groups, they agree to support its proper application in subsequent proposals. Little good comes from implementing a proprietary model if there is no internal management buy-in, or no support from the key customers on the estimating technique.

Also, the firm's management will want to understand the results of the feasibility study so it can properly assess the financial investment required to support model development and on-going maintenance activities, such as training, model enhancements, and software corrections. On receiving approval to begin development from internal and external management, the contractor establishes an implementation team to guide the creation of a valid proprietary model. This team should include representatives from the company, key customers, DCMA, and DCAA. Appendix J, Establishing a Parametric Implementation Team, provides information on assembling a joint Industry and Government team.

4.2.2 Step 2: Preliminary Model Design

Preliminary model design begins after management approval is received. During this phase, the team refines the scope of the model's requirements, and defines the methods and assumptions which establish the basis for its business rules and estimating relationships. User requirements and input/output interfaces are also identified.

4.2.3 Step 3: Information Systems Needs

When implementing a complex proprietary model, the organization should commit and obtain the necessary resources for information systems development and support activities. Information systems support is required for a variety of functions such as:

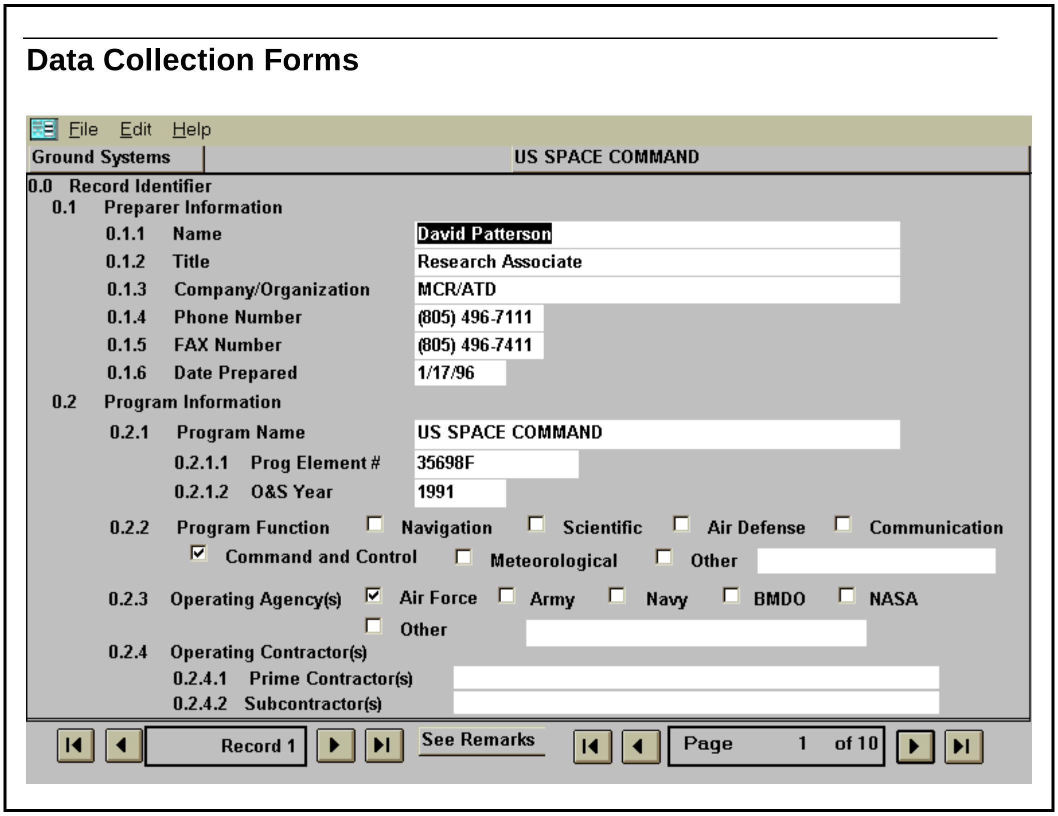
- Defining the formal system requirements needed to support the cost estimating model (e.g., hardware, software, interfaces with other systems);
- Developing the model in accordance with the company's defined methods for systems engineering (including software development);
- Testing the model to ensure it adequately satisfies all end-user requirements;
- Maintaining the integrity of the model throughout its life span by establishing procedures to manage and control all changes (i.e., configuration management);
- Providing software support services once the model is deployed to keep it operational (e.g., corrections, revisions, miscellaneous enhancements).

When an organization implements a complex proprietary model, the effort required to support software development and other activities can be extensive and should be considered in the cost-benefit analysis. When simpler models are implemented (e.g., spreadsheet models), the degree of support is smaller, but the configuration management and long-term maintenance issues still must be addressed.

4.2.4 Step 4: Data Collection and Analysis

Historical costs should be used, with the development team ensuring that they are relevant to the firm's current operating procedures. Figure 4.3 illustrates a collection form which creates consistency when recording data, maintains it digitally (e.g., in a database), and makes it easier to analyze.

In an effort to include as much relevant cost data as possible, analysts normalize it as it is incorporated into the database. They adjust data so it is as homogeneous as possible (e.g., similar in content, time value of money, quantity), and does not contain anomalies. Programmatic, non-cost data may also require normalization. The analyst must assess the condition of each program's data and make appropriate adjustments as required. Chapter 2 provides detailed information on data collection and analysis.



Data Collection Forms

File Edit Help

Ground Systems | US SPACE COMMAND

0.0 Record Identifier
0.1 Preparer Information
0.1.1 Name: David Patterson
0.1.2 Title: Research Associate
0.1.3 Company/Organization: MCR/ATD
0.1.4 Phone Number: (805) 496-7111
0.1.5 FAX Number: (805) 496-7411
0.1.6 Date Prepared: 1/17/96
0.2 Program Information
0.2.1 Program Name: US SPACE COMMAND
0.2.1.1 Prog Element #: 35698F
0.2.1.2 O&S Year: 1991
0.2.2 Program Function: ☐ Navigation ☐ Scientific ☐ Air Defense ☐ Communication ☑ Command and Control ☐ Meteorological ☐ Other
0.2.3 Operating Agency(s): ☑ Air Force ☐ Army ☐ Navy ☐ BMDO ☐ NASA ☐ Other
0.2.4 Operating Contractor(s)
0.2.4.1 Prime Contractor(s)
0.2.4.2 Subcontractor(s)

Record 1 | See Remarks | Page 1 of 10

Figure 4.3 Example of a Computerized Data Collection Form

When developing a model, the team identifies the main characteristics, called the primary cost drivers, that are responsible for, and have the greatest impact on, the product or services cost to be estimated. As many primary cost drivers as possible should be identified and included. Chapter 3, Cost Estimating Relationships, addresses the topic in more detail.

4.2.5 Step 5: Model Development

The development of a proprietary model incorporates many anticipated uses and goals such as estimating/users' requirements, availability of credible data, life-cycle costs, systems engineering costs, forward pricing rates, and it must integrate these into the parametric estimating approach. The modeling process, in particular, focuses on these tasks:

- Specifying the estimating methods for accomplishing the estimating goals;
- Identifying the job functions and other elements of cost that will be estimated;
- Defining data input structures and WBS elements.

Proprietary models may contain a number of different estimating techniques (e.g., CERs, the input of discrete estimates), and must document how they all interact.

Figure 4.4 shows some of the parametric equations used by the Space Sensor Cost Model. The model is a statistically derived aggregate of CERs based on historic data collected from national sensor programs, including DMSP, DSP, Landsat, AOA, and thirty space experiments. The CERs predict contractor cost without fee and are based on engineering cost drivers, including:

D = Detector chip area in square microns

AE = Number of active elements in the focal plane array

W = Wavelength in Microns

C = Cooling capacity in watts

I = Input power per cooling capacity

AS = Optical area sum in square centimeters

AA = Optical area average in square centimeters

ALW = Area x length x width in square centimeters

OD = Optical element dimension in centimeters

WBS Element	Development and Production CERs (CY2006K$)			
	Engineering	Prototype T1	Prod Setup	Flight Unit T1
Focal Plane Array – Monolithic	1936 (avg value)	5 + 5E-07 * D	159 (avg value)	11 + 3.75E-04 * AE
Optical Telescope Assy	854 - 1996 * W + 5.61 * AS - 9.7 * AA	253 + 1.13 * AS - 2.22 * AA	184 + 0.16 * ALW + 7.67 * OD	- 63 + 3 * AS - 5.42 * AA
Cryogenic Cooler	1028 + 510 * C	- 142 + 402 * C + 3.3 * I	8361 (avg value)	485 (avg value)

Figure 4.4 The Space Sensor Cost Model Engineering Cost Drivers

This model meets the developer's criterion of being able to fine tune the estimate, since separate CERs are available for the engineering, prototype (or qualification unit) T1, the production setup, and the flight unit (production) T1 costs. This model can also be used for engineering trade studies and as the primary method of generating a cost proposal. The CERs were heuristically derived, then calibrated to the normalized historic data.

Another model, the Follow-On Production Model, incorporates a number of estimating techniques. It estimates follow-on production costs, allows the input of discrete estimates for certain cost elements, and uses CERs to estimate others. For example, unique non-recurring data and travel costs are discretely estimated and input to the model; however, material can either be entered as a discrete estimate, or the analyst can use the model to estimate the costs through the application of an integrated material price database. The model estimates assembly hours with a "best fit" improvement curve slope extrapolated from prior build history, and can use expert judgment input to predict the point where the improvement slope flattens. Inspection hours, miscellaneous material, other direct costs, and all other support hours are based on CERs.

The Space Sensor and Follow-On Production models demonstrate that proprietary models can be designed for specific estimating needs, given carefully defined requirements.

4.2.6 Step 6: Calibration and Validation

Parametric models are calibrated and validated before they are used to develop estimates for proposals. Since proprietary models are based on an organization's historical data, they are considered to be self-calibrated. Chapter 5, Complex Hardware Models, and Chapter 6, Complex Software Models, discuss the calibration of commercial models. The validation process, however, applies to all parametric estimating techniques, whether CERs, proprietary models, or commercial models.

Validation is the process, or act, of demonstrating the proprietary model's ability to function as a credible estimating tool. Validation ensures:

- Estimating system policies and procedures are established and enforced;
- Key personnel have proper experience and are adequately trained;
- Proper information system controls are established to monitor system development and maintenance activities in order to ensure the model's continued integrity;
- The model is a good predictor of costs.

Models should be validated and periodically updated to ensure they are based on current, accurate, and complete data, and that they remain good cost predictors. A contractor should work with Government representatives to determine how frequently a proprietary model is to be updated, and this decision incorporated into the company's estimating policies and procedures. Chapter 7, Government Compliance, provides further information on this subject.

The purpose of validation is the demonstration of a model's ability to reliably predict costs. This can be done in a number of ways. For example, if a company has sufficient historical data, data points can be withheld from the model building process and then used as test points to assess the model's estimating accuracy. Unfortunately, data sets available are often extremely small, and withholding a few points from the model's development may affect the precision of its parameters. This trade-off between accuracy and testability is an issue model developers always consider.

When sufficient historical data are not available for testing, accuracy assessments can be performed using other techniques. For example, a comparison can be performed between an estimate developed from a proprietary model and one prepared using other estimating techniques, as shown in Figure 4.5.

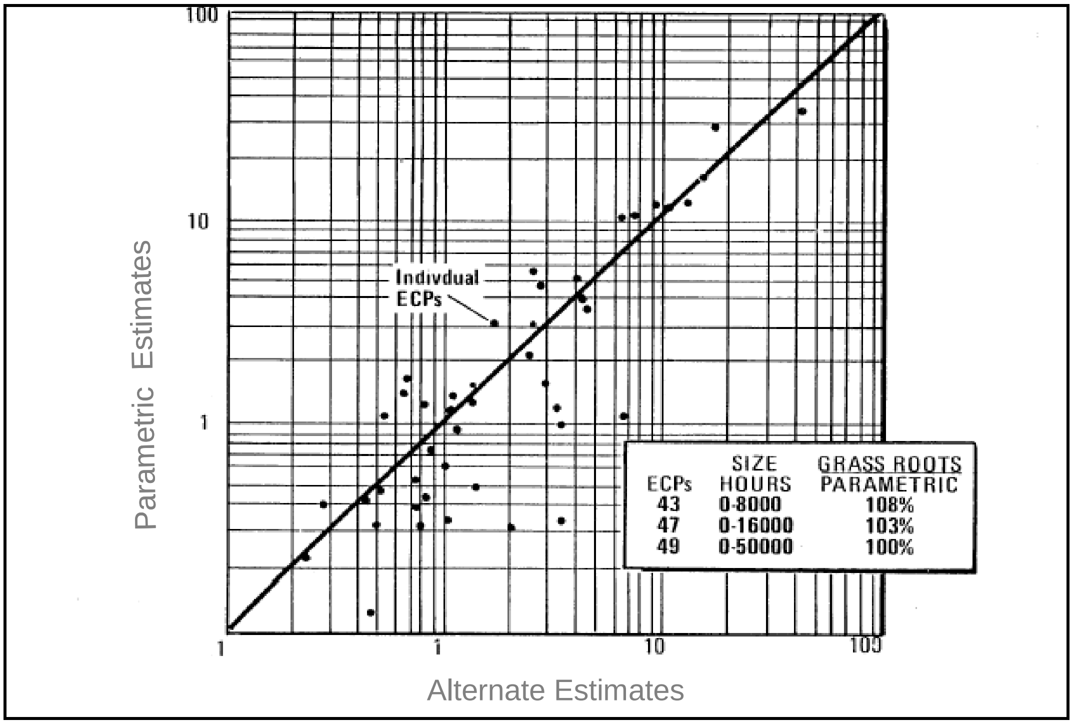


Figure 4.5 Example of Model Validation by Comparing the Model's Estimate to Another Estimate

Another testing methodology compares a program's final cost to the proprietary model's estimate of it. However, it may be months, or years, before this approach can be applied to a given program. The model team may use this method when a program is near completion, or is at a point where a meaningful earned value performance index can be determined.

Finally, a team participating in the Parametric Estimating Reinvention Laboratory developed another testing technique. It first evaluated a given model against the data used in its development to assess how well it could estimate the information from which it was built. The team then took known cost data, determined what the values of the input parameters had to be to generate these costs, and obtained help from independent experts in determining the reasonableness of those input values.

4.2.7 Step 7: Estimating System Policies and Procedures

After validation, the company must modify its estimating system policies and procedures to explain the appropriate use and application of the model for reviewers and company users. In particular, the model's developers need to document its proper use as a valid bidding tool. Chapter 7, Government Compliance, provides more information.

Companies should also explain the model's design, development, and use. For example, the contractor, as part of its support for the Follow-On Production Model and Estimating Tool, developed a detailed manual containing information about the mechanics of the model, its estimating methodologies, and the timing of updates. The company also amended its Estimating System Manual to include a section on the model, and to refer the reader to the model's own manual.

4.2.8 Step 8: Internal Approval Process

When establishing a parametric estimating implementation team, some company members may believe that gaining the Government's acceptance of a model will be more difficult than obtaining it from their own firm. In practice, however, the company's internal approval process may be equally challenging, since the development team must demonstrate to company program managers and their technical community that the model reliably estimates departmental budgets.

Model developers need to assure company representatives that the model relies on the firm's historical data and, therefore, captures how the company executed similar projects in the past. Any departmental budget allocations produced by the model should reflect the average budgetary split the firm has historically experienced. Developers should also consider the fact that a model, if approved, might change the way the company anticipates executing an existing (or planned) program (e.g., the project director may need to shift work and modify the budget). This obviously affects the circumstances under which other company personnel would approve the model.

A best practice from contractor experience involves the integration of the company representatives into the model implementation team. As an example, when implementing the Follow-On Production Model, the model designers, from the beginning, solicited the participation of key internal representatives. During the development of each module, the team incorporated the inputs of the functional department primarily responsible for executing that portion of the project which the module was designed to estimate. Although the Finance Department led the model building effort, it continuously reviewed its progress with representatives from the Engineering and Manufacturing Departments. These representatives were responsible for coordinating and obtaining any necessary information from their organization and keeping management informed.

4.2.9 Step 9: External Approval Process

Although a company may internally approve a model, the customer must also be shown that the estimating approach is valid. The Parametric Estimating Reinvention Laboratory demonstrated that involving customers in up-front decision facilitates their acceptance of parametric techniques (see Appendix J, Establishing a Parametric Implementation Team). In addition, since a customer generally receives and accepts recommendations from the local DCMA and DCAA on issues related to a contractor's parametric estimating system, it is important to include representatives from those organizations on the implementation team. Failure to do this makes it difficult and risky for the company to use a model on a proposal. The following examples provide approaches that several contractors found helpful in implementing proprietary models, and presenting them to the buying organizations, DCMA and DCAA, and prime contractors (in the case of subcontractor estimates).

In seeking acceptance of the Program Management (E-PROMM) Model, the company formed a Continuous Improvement Process (CIP) team. The team's composition included company representatives from various departments, DCMA, and DCAA. All team members participated in establishing selection criteria for the model's database. Based on the selection criteria, the contractor personnel collected actual cost data from over 40 contracts. DCMA and DCAA reviewed the data for accuracy. At the end of the data collection and model evaluation period, the DCMA and DCAA accepted the model for use in proposals to the Government. When using the model for the first time with a buying organization, the CIP team invites the buying organization to the company for a joint review and explanation of the model.

The company team developing the Space Sensor Model pursued external acceptance in a similar manner. Immediately after obtaining funding to develop the model, the developing company discussed it with other contractors, additional government organizations, and the Federally Funded Research and Development Center (FFRDC) to ensure widespread support in data collection and model validation. The model's sponsor also formed an integrated product team (IPT) to provide visibility into the cost estimating process, and to involve these groups in technical, process, and business/regulatory decisions. The IPT enabled the Government to provide real-time feedback, and guided the contractor in implementing a cost model acceptable to the Government and other contractors, and as a BOE. This IPT philosophy also improved Government understanding of the data in the model, how the model works, and how contractors intended to employ it.

Including customers on the development team does not guarantee a model's acceptance, of course. It does ensure that the customer has a voice in the model's design and usage, but the model's ability to reasonably predict costs is the ultimate basis for acceptance. No person, internal or external to the company, can prove this before final development and testing.

4.2.10 Step 10: Model Maintenance

Through the development process, the team develops a sense of how often the model needs updating. Maintenance activities include not only the incorporation of new data into the model, but also an evaluation of the mathematical relationships between the technical parameters and the costs the model estimates. Periodic evaluation of the model is required to ensure the estimates are relevant and the contractor is using the most current, accurate, and complete data, as required by the Truth in Negotiations Act (TINA). Chapter 7, Government

Compliance, discusses TINA requirements. The following examples show how two contractors maintain their proprietary models.

For the Follow-On Production Model, the Pricing Organization was identified as the office responsible for maintaining the model. The organization annually updates all cost data. The model is updated following the completion of a relevant program, or after identifying a substantive change to a relevant program if that has a significant impact on cost allocations. The Purchasing, Manufacturing, and Engineering departments work closely with Pricing to keep the organization informed of any technical additions or modifications to the model's data or algorithms.

For the Space Communications Payload Model and the Space Sensor Model, the Engineering Operations department of the company has maintenance responsibility. The programs and proposal activities that use the model provide maintenance funding. New data are contributed as programs mature and, occasionally, from non-company sources. In some situations, the cost modelers develop new CERs, based on a subset of the original database, to better match a new estimating requirement.

The process of maintaining a model involves keeping an audit trail of the CERs developed, the data points used, and their statistical effectiveness. Figure 4.6 illustrates a method for documenting company developed models, one which identifies all dependent and independent variables, CER statistics, and data points.

GRP_AT10=0.4+0.4*T_A_WTIN+23.6*LRU_MOD

Where: GRP_AT10 = X, in BY97K$.
and: T_A_WTIN =
LRU_MOD =

CORRELATION MATRIX:

	GRP_AT10	T_A_WTIN	LRU_MOD
GRP_AT10	1.00	0.48	0.78
T_A_WTIN	0.48	1.00	-0.13
LRU_MOD	0.78	-0.13	1.00

CER FIT STATISTICS:

# of OBS	MEAN	SEE	CV	ADJ R^2	R^2
9	293.1	39.7	13.6%	0.93	0.95

INDEPENDENT VARIABLE INFORMATION:

VARIABLE	T-STAT	STD ERR	SIG-T	BETA COEFF	RANGE	MEAN
CONSTANT	0.0	30.5	0.99	N/A	N/A	N/A
T_A_WTIN	6.4	0.1	0.0	0.59		256.53
LRU_MOD	9.4	2.5	0.0	0.86		7.89

ANOVA TABLE:

SOURCE	Sum of Squares	DF	Mean Square Error	F Ratio/SIG (F)
Regression	182481.5	2	91240.75	57/0.0
Residual	9474.6	6	1579.09	
TOTAL	191956.1			

DATA POINTS:

REC-1
REC-2
REC-3
REC-4
REC-6
REC-7
REC-8
REC-9
REC-10

Figure 4.6 Example of Model Documentation Which Facilitates Maintenance

4.3 Evaluation Criteria

An evaluator's review of a proprietary model generally focuses on determining that:

- Policies and procedures exist which enforce the appropriate use of, and consistency in, the model;
- Data used to develop the model are credible and verifiable;

- The proprietary model has been validated for use as a BOE.

Chapter 2, Data Collection and Analysis, and Chapter 3, Cost Estimating Relationships, contain general evaluation criteria for these elements. Chapter 7, Government Compliance, provides Government evaluation criteria. There are two additional areas that evaluators should consider reviewing: the cost benefit analysis, and information system controls.

4.3.1 Cost Benefit Analysis

A company should perform a cost benefit analysis to determine whether a proprietary model's expected benefits outweigh the costs to implement and maintain it. Items that should be considered include:

- The cost to develop and maintain the proprietary model;
- Frequency of use;
- Expected savings;
- Customer support.

As a best practice, companies should consider the return on investment when implementing a new parametric technique.

4.3.2 Information System Controls

Information system controls make certain a model is economical, efficient, and that it executes management policies in a controlled environment. Some key issues are:

- Does the system documentation thoroughly describe the model and include:
    - Processing performed by the model;
    - Data processed by the model;
    - Reports generated by the model;
    - User instructions.
- Assurance that proper controls are established to monitor changes to the model.
- Assurance that proper security controls are established and updated on a regular basis.
- Trained and experienced people perform model development and maintenance.
- Testing was performed to ensure the model functions properly.

The effort needed to evaluate information system controls will vary with the complexity of the model. The purpose of the controls is to maintain the model's integrity.

4.4 Summary Example: Jet Engine Cost Model

This complex model, also referred to as the Cost Offering Method for Affordable Propulsion Engineering Acquisition and Test (COMPEAT$™) was designed for multiple uses, including the estimation of life cycle costs for military engine development proposals, commercial engine studies, and target costing. COMPEAT$™ includes six modules: engineering, engine test, systems engineering, engine hardware, component test, and operations and support.

In COMPEAT$™, general part characteristics or key cost drivers are used as the model's inputs such as engine size, part features, dimensions, and efficiency. The model compares these inputs to a database of historical parts, and selects the best historical match using a hierarchy of criteria. Based on the degree of similarity between the new and matched parts, the model processes the new part's characteristics through a series of CERs (themselves based on historical part characteristics and known costs) to produce a cost estimate.

To support the COMPEAT$™ Model, the contractor negotiated an Advance Agreement with its Government customer. This agreement was used to define the model maintenance requirements (i.e., frequency of updates) based on TINA provisions. For example, the Agreement established the boundaries for use of the model in Government proposals, and addressed the model's data, data files, training of users, and the timing of updates. The company later incorporated the information from this agreement into its standard estimating system policies and procedures and its Cost Estimating Manual.

The company team developing the COMPEAT$™ Model pursued external acceptance in a similar manner. Immediately after obtaining internal company support, the company initiated dialogue with its local Government representatives. Integrating the Government and company personnel early in the process promoted a "no surprises" philosophy.

The company's Engineering Operations organization has maintenance responsibility for COMPEAT$™. The Military Engine, Commercial Engine, Engineering, and Manufacturing Operations fund its maintenance costs. While the Engineering organization updates the development data and all the model algorithms, Aircraft Engines Systems personnel gather the hardware data. The Proposals group evaluates and screens all data, particularly cost, for reasonableness.

4.5 Lessons Learned

The Parametric Estimating Reinvention Laboratory identified some concepts that all implementation teams should consider:

- No company or individual can develop a valid model without the participation of a number of key people;
- Include the customer, all interested company personnel, and DCMA and DCAA representatives;
- Establish a process flow and target development dates to ensure all team members provide their inputs to the model's design;
- Consider the costs and benefits of model development;
- Evaluate commercial models as an alternative to proprietary development;
- Remember that the goal is to establish a more efficient and reliable estimating system, not just create a model.

4.6 Best Practices

Based on Parametric Estimating Reinvention Laboratory experience, no single implementation approach is superior to another, but all successful applications of the general model-building process do depend on good communications. Because Industry and Government recognize a common need to reduce the time and expense of generating, evaluating, and negotiating cost proposals, they agree to participate on a particular model implementation team.

Industry model team members provide the Government insight into the methods and constraints of their estimating processes, and the Government team members explain what criteria the model must meet for it to be an acceptable estimating tool. As the work progresses, all team members share opinions, concerns, and solutions in an effort to make the proposal preparation process faster and less costly, while maintaining a reasonable level of reliability.

The best practices for model development are:

- Obtain internal and external senior management sponsorship of the initiative early in the process.
- Estimate and track the cost of developing and implementing the new methods. Maintain metrics on cycle times and proposal costs to determine the return on the invested costs.
- Engage the major customers, DCMA, and DCAA early in the process and solicit their input on a real-time basis.
- Engage the company functional communities early in the process to ensure that they can manage with the outputs the model provides.
- Use cross-functional, Government/contractor IPTs to facilitate model development and acceptance.
- Develop and rely on a process similar to that illustrated in Figure 4.2.

CHAPTER 5
Complex Hardware Models

This chapter provides an overview of complex (in-house, commercially available, and Government developed) hardware parametric models, and describes the recommended processes for using them to develop estimates for a wide range of applications and decision support including government project office estimates, and proposals. When properly implemented and appropriately used, parametric estimating models reliably predict future project costs more efficiently than traditional estimating methods. It is not intended that this chapter cover every hardware model, but provide a general description of the generic process.

The chapter includes:

- General information on the parametric cost modeling process for hardware models;
- Recommended estimating system policies and procedures for implementing and using hardware models;
- Processes for using models as a basis of estimate (BOE) on proposals submitted to the Government or higher tier contractors;
- Best practices and lessons learned from the Parametric Estimating Reinvention Laboratory.

The chapter provides best practice recommendations which are based on model practitioner's experiences with implementing hardware models into an organization's cost estimating and analysis practices. Many models are available, and some are used for very specific purposes such as estimating the costs of electronic modules and the operations and support cost of hardware and software systems. Organizations and cost estimating model users are encouraged to evaluate as many alternatives as possible prior to selecting and implementing the most appropriate cost estimating model that meets their requirements.

5.1 Background

In the early 1950's, the Rand Corporation pioneered parametric cost estimating concepts and used them to develop costs in support of high-level planning studies for the United States Air Force (USAF). Rand used parametric cost estimating relationships (CERs) based on speed, range, altitude, and other design parameters for first and second-generation intercontinental ballistic missiles, jet fighters, jet bombers, and cargo aircraft.

Since then the Government and Industry cost analysis community has moved from simple to complex CERs, and then to sophisticated computer parametric models that can estimate the life-cycle cost (LCC) of complex weapon, space, and software-intensive systems. A parametric cost model can be viewed as the collection of databases, CERs (simple one-variable equations and complex algorithms requiring multiple design/performance/programmatic parameters), cost factors, algorithms, and the associated logic, which together are used to estimate the costs of a system and its components. A model may be manual or automated and interactive. A parametric cost model uses known values (e.g., system descriptions or parameters) to estimate unknown ones (e.g., program, component, activity costs).

Over the past 40 years, Government and Industry have used parametric models to support conceptual estimating, design-to-cost analyses, LCC estimates, independent cost estimates, risk analyses, budget planning and analyses, should cost assessments, and proposal evaluations. Chapter 8, Other Parametric Applications, contains information on other uses of parametric models.

In 1975, the then RCA Company offered a commercial hardware estimating model, which was initially developed in the 1960s to support internal independent cost estimates. This tool, and others that followed from competing companies, grew in popularity and sophistication and were used to support the full spectrum of cost estimating and analysis activities. However, these hardware cost models were generally not used as a BOE for proposals submitted to the Government when cost or pricing data were required.

As part of the recent Parametric Estimating Reinvention Laboratory effort, several companies using integrated product teams (IPTs) implemented commercial parametric estimating hardware models, which can rapidly compute development and design costs, manufacturing costs of prototypes, and production unit/manufacturing support costs. The models can also compute the operation and support costs of fielded systems.

5.2 Overview of Hardware Cost Modeling

Hardware cost models provide estimates of system acquisition costs, schedule, and risks based upon:

- Quantitative parameters such as complexity, quantity, weight, and size;
- Qualitative parameters such as environmental specifications, type of packaging, and level of integration;
- Schedule parameters such as months to first prototype, manufacturing rate, and amount of new design.

Hardware parametric models bring speed, accuracy, and flexibility to the cost estimating process. Cost measurement of alternative design concepts early in the design and acquisition process is crucial to a new program because there is little opportunity to change program costs significantly once a detailed design and specs have been released to production. The analyst, with engineering support, reviews the system's concept of operations, system requirements, documentation, and conceptual designs. From this review, a work breakdown structure (WBS) and cost element structure (CES) are developed for all the systems that are being designed, developed, and produced. In addition, ground rules and assumptions (GR&As) defining the acquisition drivers and the programmatic constraints that affect design and performance are identified. This WBS/CES is then incorporated into the model, and it defines what is being estimated, including the descriptive parameters.

Parametric estimating models have been developed to operate with limited concept description so that program management personnel can estimate the cost of many unique configurations before system design specifications and detailed bills of material are finalized. Parametric models can also be used as the basis of a cost estimate in preparation of firm business proposals, or in the independent assessment of cost estimates prepared using a traditional estimating approach.

Hardware models extrapolate from past systems to estimate and predict the costs of future ones, and their inputs cover a wide range of system features and characteristics. Weight and size are often used as a model's principal descriptive variables (descriptors) since all systems (and their components) exhibit these properties. Application and type are the common predictive variables (predictors) for electronic components, while mechanical and structural elements can best be described in terms of their construction: method, type of material, functionality, machinability, and manufacturing process.

Some uses of parametric hardware cost models include (see Chapter 8 for more discussion on cost models):

- Cost/price proposal preparation;
- Evaluation of design alternatives and procurement and acquisition options;
- Cost realism analysis;
- Cost and schedule risks;
- Estimates of cost to complete;
- Estimates of modifications;
- Should cost analysis;
- Most probable cost estimates;
- Evaluations of bids and proposals (sanity checks);
- Vendor negotiations;

- LCC estimates;
- Independent cost estimates;
- "What if" design cost trade-off analysis or Design-to-Cost (DTC);
- Bid/no bid decisions;
- Estimates of spare parts costs and other operations and support (O&S) costs.

Parametric models can be used in all phases of hardware acquisition; for example, development, production and deployment, and all functional aspects such as purchased and furnished hardware (GFE), hardware modifications, subcontractor liaison, hardware-software integration, multiple lot production, and hardware integration and test.

Figure 5.1 depicts typical hardware modeling inputs and outputs. The main advantage of a parametric model over grass roots or build-up methods is that it requires much less data to make the estimate. For example, when a parametric model calculates a manufacturing cost, it does so using a few items of programmatic, technical, and schedule information rather than an itemized parts list and/or a labor resources build-up.

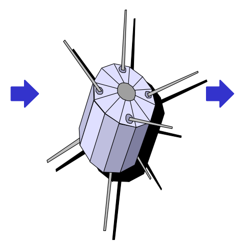
Fundamental input parameters for parametric hardware models include:

- Functional design parameters;
- Quantities of equipment to be developed, produced, modified, subcontracted, and integrated and tested;
- Applications (technology of materials and processes) of structural and electronic portions of the hardware;
- Hardware geometry consisting of size, weight of electronic and structural elements, and electronic packaging density;
- Amount of new design required and complexity of the development engineering task;
- Operational environment and specification requirements of the hardware;
- Schedules for development, production, procurement, modification, and integration and testing;
- Fabrication process to be used for production;
- Yield considerations for hardware development;
- Pertinent escalation rates and mark-ups for general and administrative charges, profit, cost of money, and purchased item handling.

Hardware Modeling

A Hardware Model Uses Common Parameters To Estimate and Evaluate New Requirements

- Input Parameters
  - Magnitude (quantity)
  - Operating environment
  - Amount of new design & design repeat
  - Engineering complexity
  - Manufacturing complexity
  - Schedule
  - H/W - S/W integration
  - Weight / volume



- Output Parameters
  - Cost
    - Development
    - Production
    - Engineering
    - Manufacturing
  - Schedule risks
  - Unit /system integration costs

Figure 5.1 Hardware Model Input and Output Parameters

The fundamental feature of parametric inputs is their inter-relationship with the other elements of the WBS/CES. An effect caused by a change in any one parameter does not usually impact just one cost element, but several. For example, consider the impact of a change in quantity. It certainly affects the manufacturing and spares cost (specifically, cost per unit time), and can also affect the fabrication process and also, in a ripple effect, the cost of tooling and test equipment. The same change in quantity could alter the production schedule, which changes the costs associated with escalation, integration and test, sustaining engineering, and project management. This interaction is characteristic of and captured consistently in most parametric input variables and models.

A model's input parameters uniquely define the hardware configuration (what is being estimated and modeled) used for cost estimating and modeling. The resulting cost output is determined by the model's mathematical relationships, algorithms and data. As stated, cost may be estimated with a minimal number of inputs. It is always preferable, however, to obtain as much information as possible to be incorporated into the parametric model by working with the designers and engineers to define the appropriate inputs, since doing so will reduce the statistical uncertainty associated with the input variables.

Finally, a comprehensive parametric model has the capability to:

- Incorporate a WBS/CES and, define, characterize and quantify the logical variables;
- Be calibrated;
- Estimate the cost of multiple lot production;

- Calculate manufacturing costs of non-homogenous assemblies;
- Determine the cost impact of compressing or extending development or production schedules;
- Estimate the cost impact of the development schedule (concurrency or lapse) on production;
- Perform cost and schedule risk analysis.

5.3 The Parametric Cost Modeling Process

Figure 5.2 shows the major steps in the cost modeling process. This section discusses them in turn.

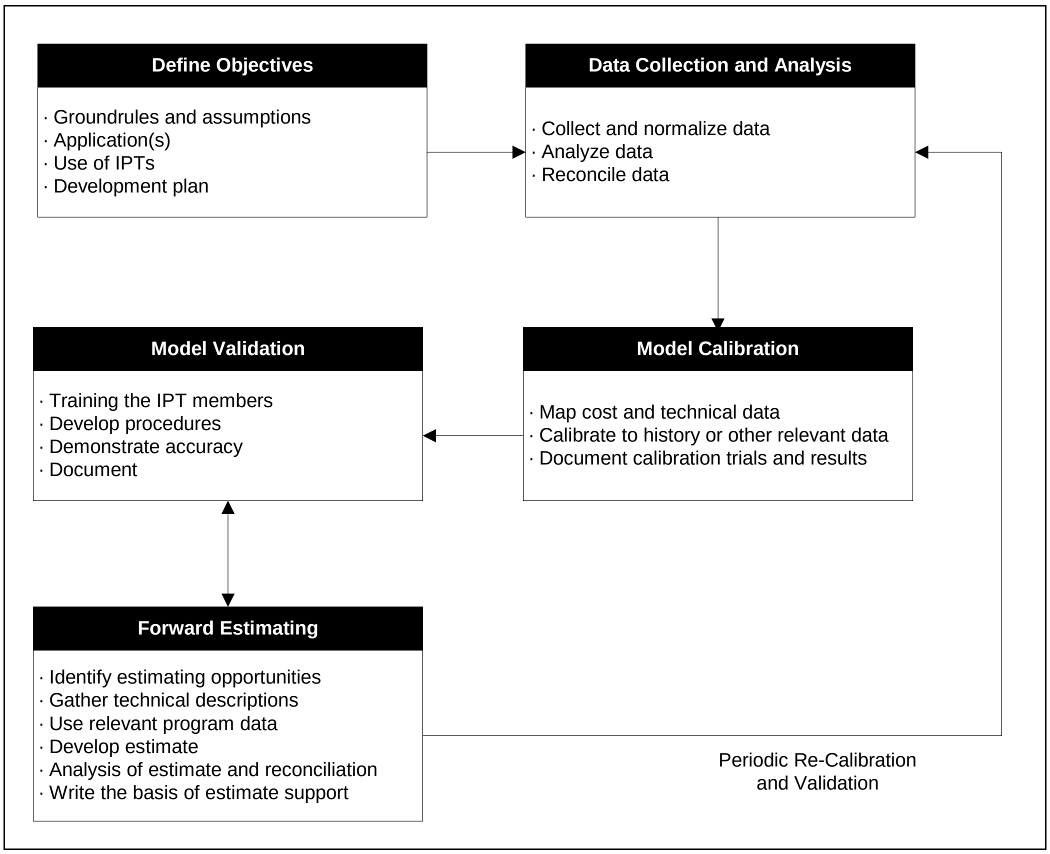


Figure 5.2 Complex Parametric Hardware Model Estimating Process

5.3.1 Define Objectives

Users of complex hardware models must first establish assumptions concerning data collection, data requirements for model calibration/validation, and the best way to normalize data for differences in development, production quantities, scope of work. This includes establishing ground rules for determining the compatibility of the data, the model itself, the calibration results, and the proposed use of the model. The estimator should coordinate these rules and assumptions with the proposal manager, technical leads, customer, DCMA, and DCAA.

5.3.2 Data Collection and Analysis

Historical cost and labor hour data are required as a basis for the model's cost estimate, as well as technical non-cost data which describes the physical, performance, and engineering characteristics of a system or component to be estimated. The collecting point for company-specific cost and labor hour data is, in most cases, the company's general ledger or it's ERP, or other information system.

Data should be collected and maintained with an audit trail. All financial data used to calibrate complex cost models must be consistent with and traceable back to their sources. Historical cost data may need to be normalized or adjusted to account for differences related to scope of work, program anomalies, changes in technology, new business practices, inflation, learning curve and quantities, and production rate.

Technical data comes from a variety of sources including engineering drawings, specifications, mass properties (i.e., weights), preliminary and critical design review documents, suppliers, and other engineering and manufacturing records. Schedules and quantities come from a variety of sources, including the planning department, industrial engineering, procurement files, and business management.

The Cost Analysis Requirements Document (CARD) for a DoD program often contains much of the information needed for a model. Chapter 2, Data Collection and Analysis, discusses this in more detail.

One of the most important aspects of data collection is interviewing the technical personnel involved in the design, analysis, manufacturing, assembly, and test of existing and planned hardware development and production programs. This provides an opportunity to gather data on issues such as heritage, degrees of complexity, inheritance, and new business initiatives. The interview also helps the analyst compare the degree of complexity of the projects within the model's database to that being estimated. The analyst then compiles all the parametric data that has been collected, and stores it in a format that the model can use.

5.3.3 Model Calibration

The calibration of a complex hardware model is the process of tuning it to reflect the given contractor's historical cost experience and business culture. Actual technical, programmatic, and cost data from previous projects embody the organization's historical way of doing business. The parameters may have to be adjusted for the way an organization's business will be conducted in the future. Calibration captures this by adjusting the complex model's complexity and/or adjustment factors. The calibration process involves:

- Collecting cost, technical, and programmatic data from historical or on-going relevant programs;
- Analyzing, reconciling, and mapping cost accounts to complex model cost element terminology;
- Using the model to calculate complexity values (e.g., manufacturing complexity, material index, drafting and design global values, calibration adjustment factors, and assessing the effects between the current and future processes and procedures).

Calibration results are documented in a program notebook, together with all ground rules and assumptions, summary input data, technical descriptions, and calibration output reports. Any adjustments made to the model's parameters are included, along with the associated rationale. The estimator should follow the calibration process recommended by the model's supplier. Figure 5.3 illustrates the calibration process for a commercial model. A similar process is used to calibrate other complex models.

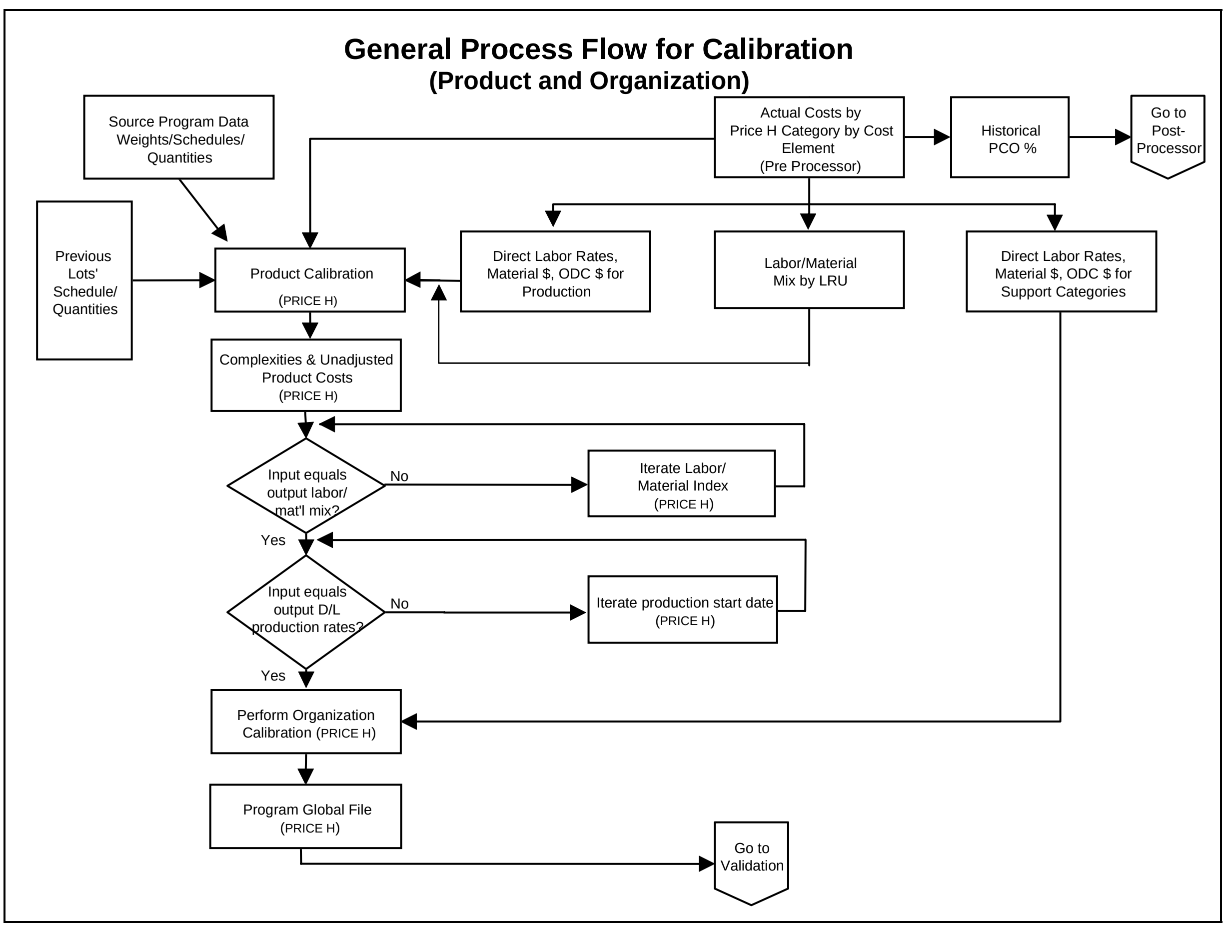


Figure 5.3 Detailed Calibration Process Flow

As an example (from the Parametric Estimating Reinvention Laboratory), Lockheed Martin Astronautics (LMA) provided the following information on the process it uses to calibrate PRICE H®. LMA has an automated Estimating Database (EDB) that allows users to access designated historical cost data on all programs, by month, since 1985.

The EDB is the basis for LMA's successful implementation of parametric estimating techniques. Even with this valuable resource, extensive analysis and mapping is required to enter this cost accounting history into PRICE H®. For instance, LMA's design and analysis activities correlate with the PRICE H® categories of drafting and design.

LMA's major effort in data collection and analysis consists of gathering technical data and performing interviews with Product Integrity Engineers to characterize the hardware subsystems relative to functionality, experience, new design, heritage, and engineering and manufacturing complexities. Data normalization consists of establishing ground rules on what types and classes of costs will be used for calibration. For instance, in LMA's calibration process, travel and other direct costs are excluded from the model because they are estimated using other techniques.

The following list summarizes the calibration lessons learned, best practices, and examples from the Parametric Estimating Reinvention Laboratory sites that implemented complex hardware models.

- Calibration can be a resource intensive task. A majority of the effort will be expended on data collection, interviewing, analysis, and reconciliation of technical, cost, and programmatic data.
- Ensure full documentation, including ground rules and assumptions. Separate documentation of the calibration effort for each program's data is strongly encouraged. Documentation should include the interview sheets of the technical personnel, brief technical description of the program, input values used, and the actual calibration runs.
- Decide up front whether to estimate labor only or both labor and procurement (i.e., material and subcontracts).
- Ensure all end item weights align with the costs.
- Calibrate at the lowest appropriate WBS level with available cost accounting data.
- It is critical to ensure proper mapping from a company's cost accounting data to the cost element definitions used by the complex model.
- It may be necessary to re-calibrate as more data become available.

5.3.4 Model Validation

Validation is the process of demonstrating the credibility of a parametric model as a good predictor of costs, and must be done before the model can be used as the

BOE for proposals. Parametric models also require periodic re-calibration and validation of company-indexed complexity factors.

A parametric model should demonstrate the following features during its validation.

- Assurance that model users have sufficient experience as well as training (especially from the model's developer).
- Documented calibration results.
- Evidence of formal estimating procedures/practices that ensure consistency in calibrating/using a complex parametric hardware model, focusing on these areas:
  - Model background/history;
  - Listing of the key cost driver input parameters;
  - Recommended steps for calibration;
  - Recommended steps for developing an estimate;
  - Guidance for supporting the BOE in a proposal.

Many analysts use one of the following methods to assess the model's predictive accuracy.

- Predict the cost of end items not previously calibrated (using appropriate calibration values), and compare the model's estimates to the end items' actual costs or estimates-at-completion (when at least 80 percent of program actual costs are known).
- Compare the model's estimates with independent project estimates made using traditional estimating techniques.
- Compare the model's estimates to prior production estimates or negotiations.

5.3.5 Forward Estimating

Figure 5.4 displays the forward estimating process. All the collected historical complexity factors, technical descriptors, programmatic data, and interview results are used to develop the proposal estimate. The BOE should document major input parameter values and their rationales. Some companies may ask the functional areas (e.g., engineering, quality) to develop independent estimates as sanity checks to gain confidence in the complex model's results. In addition, the establishment of a reconciliation process is strongly recommended to provide a mechanism for comparing the model's estimates with actual cost experience (and can also be used for the periodic revalidation).

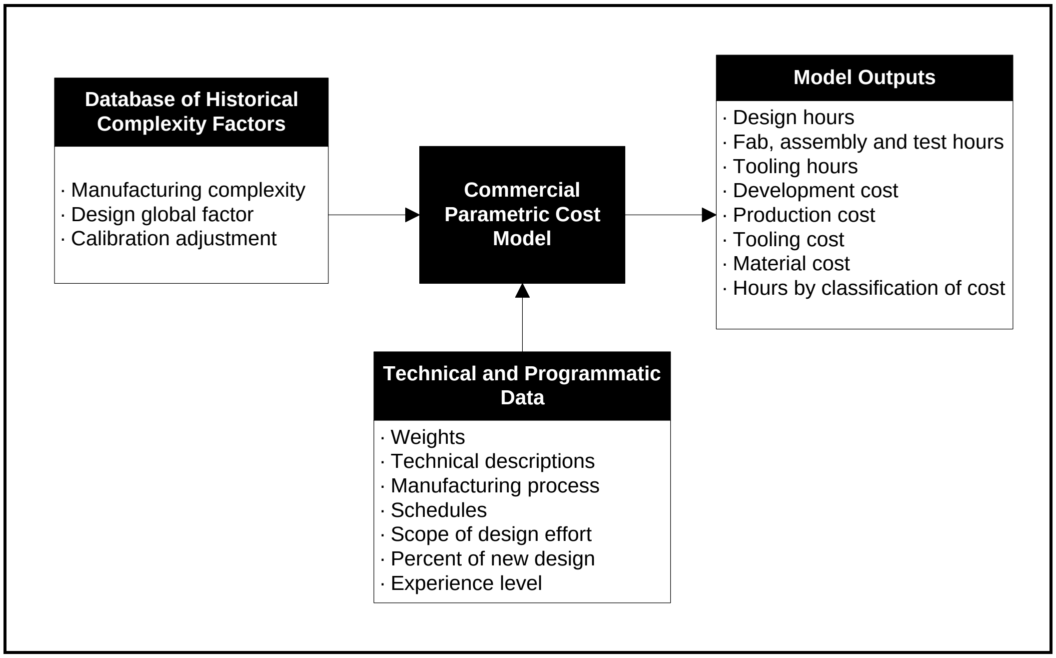


Figure 5.4 Forward Estimating Process

A model's results may not be in the usual proposal format (e.g., spread of hours by functional category and element of cost) that differs with the model and how the company chooses to use it. In this case, a post processor can restructure the results to have the desired level of detail (e.g., percentage spread of hours within a functional category). To produce a dollar estimate for the project, then the company just applies current labor and indirect rates to the post processor output to produce the typical functional category and element of a cost proposal. Chapter 7, Government Compliance, contains additional information on formats for proposal submissions. There is a trend in industry to use hardware datasheets, initially developed by the National Reconnaissance Office (NRO) Cost Group to capture technical, detailed parametric model inputs. Appendix K includes data input forms that are often requested by the NRO and other Government contracting organizations. These data input forms reflect required inputs for parametric models.

5.4 Commercially Available Hardware Models

Appendix A describes various commercial hardware models, and provides an overview of their major input variables, key cost drivers, and significant CERs.

An analyst intending to use any of these models for developing estimates, conducting sensitivity tests, or evaluating key input parameter values should take the formal training provided by the model's supplier. Additional information, including contact information, is provided in Appendix A.

5.5 Lessons Learned from the Use of Complex Hardware Models

The main lessons learned from practitioners who used and tested complex hardware models during the Parametric Estimating Reinvention Laboratory effort are as follows

- The effective implementation of complex hardware parametric models may require that a contractor's executive business/project managers support changes in the company's current business operations and processes.
- Setting specific implementation goals and rewards (e.g., 15 percent of all new business proposals will utilize complex models to develop estimates), instead of vague statements such as "encouragement to explore opportunities," increases the actual use of parametric models.
- Contractor management should ensure that enough resources are dedicated to the use of complex models since there are sizable start up costs for collecting and analyzing data, interviewing technical personnel, testing optional calibration approaches, developing a procedure for consistent calibration and forward estimating, and finalizing the overall methodology for proposal application.
- Start small to gain experience and acceptance by all internal and external customers (e.g., use simple CERs to parametrically estimate one to three subsystems in a proposal) and to demonstrate the reasonableness of proposal parametric estimates by comparing them with estimates made using other techniques.
- A company should have a champion who can continuously market the advantages of parametric estimating as a BOE tool to product area Vice Presidents and Directors. Awareness training is required for internal functions (e.g., systems engineering, mechanical and electronic design) and internal and external customers (e.g., DCAA, DCMA, Procurement Contracting Officers, price/cost analysts, technical evaluators, buying agency program managers).
- Formal commercial model training should be required for anyone attempting to calibrate historical data, or who is responsible for developing a proposal estimate requiring cost, pricing, or cost-realism data. Training also is beneficial for DCAA, DCMA, and program office personnel as well as anyone involved in the proposal evaluation process. The Parametric Estimating Reinvention Laboratory identified joint IPT training and positive team interaction as a best practice. The early involvement of all interested parties in an IPT is essential.
- The successful implementation of parametric techniques requires Contractor and Government team members to be open minded and willing to change the existing business culture. A best practice is to get customer buy-in up front, which should include those technical analysts who

evaluate proposals. Appendix J, Establishing a Parametric Implementation Team, provides additional information.

- The entire parametric model submission, review, and approval process should be defined up front.
- Establish clear guidelines for applying parametric models. This includes the range of cost outcomes, the type of effort being estimated, and conditions under which the work will be performed. Specific attention should be given to model calibration data in establishing these parameters.
- Guidance should be established for subcontractors' estimates. The optimal situation, when cost or pricing data are required, is to have suppliers develop their own estimates based on acceptable calibrated parametric models. Chapter 7, Government Compliance, discusses subcontract requirements.
- There are many ways to implement properly calibrated and validated models. Look beyond "We have always done it this way."
- The parametric estimating process will undergo continuous refinement and improvement.

5.6 Best Practices

The best practices from the Parametric Estimating Reinvention Laboratory sites where complex hardware models were used as the BOE in a proposal are as follows.

- Pursue the application of parametric estimating methods in small steps. Preparing an entire proposal for a major project as the organization's first attempt may overwhelm both the organization and the customer. One recommended approach is to initially estimate either one subsystem or component to gain the confidence of both internal and external parties.
- Involve the customer in establishing ground rules for acceptance of the model's outputs.
- Develop an independent, discrete project estimate as a secondary methodology to help establish the realism of the parametric technique.
- Include the customer in the tool selection decision.
- Develop a Parametric IPT (include customers) and train all key members in the use of the selected model.
- Include the customer and oversight groups in model calibration exercises.
- Formally document all calibration efforts (the company's Estimating Manual should contain the guidance for this documentation).

5.7 Conclusions

Complex parametric cost models provide opportunities for Industry and Government to save time and money on proposals and negotiations requiring cost or pricing data. In addition, experience from Parametric Estimating Reinvention Laboratory sites indicates that the use of these models, when properly calibrated/validated and appropriately applied, complies with Government procurement regulations.

CHAPTER 6
Complex Software Models

This chapter provides an overview of complex (in-house, commercially available and Government developed) software parametric models, and describes the recommended processes for using them to develop estimates for a wide range of applications and decision support including government project office estimates, and proposals. As in the previous chapter, "complex model" means any parametric model that uses more than one CER in the assessment of cost.

Because software spending in DoD and NASA is significant and continues to increase, it is critical for those involved in software acquisition to understand the factors that drive software development and maintenance (support) activities and costs. This chapter discusses software estimating methodologies with emphasis on parametric models used in Industry and Government and highlights common software process improvement activities that are relevant to software parametric estimating practices.

The chapter also:

- Provides an overview of the software life cycle, including different methods related to software development and support activities;
- Examines Industry and Government software process improvement initiatives;
- Discusses the software estimating process and explains different types of software estimating techniques, including parametric models;
- Explains techniques used to estimate software size (concentrating on parametric applications);
- Identifies future developments that affect software estimating;
- Discusses best practices and lessons learned from the Parametric Estimating Reinvention Laboratory.

6.1 Background

Software is a combination of computer instructions and data definitions that are required for computer hardware to perform computational or control functions. DoD spending for software intensive systems is significant and continues to increase. Software costs as a percentage of total program and computer system costs are also increasing. DoD purchases software for weapon systems and management information systems (MISs). Weapon system software is associated with the operations of aircraft; ships; tanks; tactical and strategic missiles; smart munitions; space launch and space-based systems; command and control (C^2); command, control, communications (C^3); and intelligence (C^3I) systems. MIS software also performs activities that support weapon systems (e.g., payroll and personnel, spares calculations).

Accurately projecting and tracking software costs is difficult, and cost overruns often occur. It is very important, therefore, to understand software estimating processes and methods. Software estimating problems often occur because of the:

- Inability to accurately size a software project;
- Inability to accurately specify an appropriate software development and support environment;
- Improper assessment of staffing levels and skills;
- Lack of well-defined requirements for the software activity being estimated.

Figure 6.1 illustrates the critical elements of the software estimating process, and shows that adequate parametric software estimating practices include policies and procedures for data collection and normalization, as well as calibration and validation (including guidance on significant model cost drivers, input and output parameters, and steps for validating the model's accuracy).

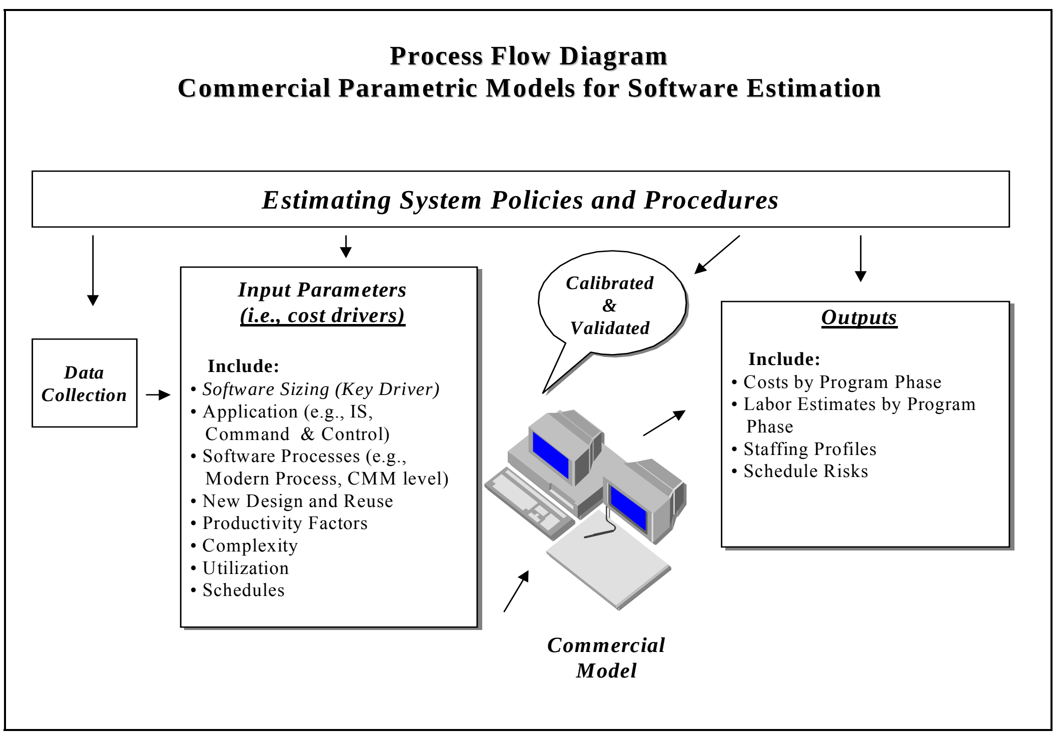


Figure 6.1 Critical Software Estimating Process Elements

Processes related to data collection and normalization, calibration, and validation are discussed in Chapter 2, Data Collection and Analysis, Chapter 5, Complex Hardware Models, and Chapter 7, Government Compliance. The USAF Software Technology Support Center's reference book *Guidelines for Successful Acquisition and Management of Software Intensive Systems (GSAM)*, updated in May 2000, provides detailed information on software estimating processes, software life cycles, process improvements, and other related information. This is a good reference for analysts who want to understand software related activities.

There is a trend in Industry to use software datasheets, initially developed by the National Reconnaissance Office (NRO) Cost Group to capture parametric model inputs of technical detail and assumptions. These datasheets are discussed in Appendix K.

Here are a few "rules-of-thumb" from the NASA *Cost Estimating Handbook*.

- Fifty five percent of software projects exceed budget by at least 90 percent. Software projects at large companies are not completed 91 percent of the time. Of the projects that are completed, only 42 percent of them have all the originally proposed features [Remer, 1998].
- Historical cost estimates for NASA projects are under-estimated by a factor of at least 2. The actual cost versus estimated cost ratio ranges from 2.1 to 2.5 [Remer, 1998]. At JPL software development cost growth is 50 percent on average from PDR [Hihn and Habib-agahi, May 2000, Hihn and Habib-agahi, Sept. 2000]
- Cost estimation accuracy using ratio estimating by phases without detailed engineering data gives an accuracy ranging from .3 percent to +50 percent. Using flow diagram layouts, interface details, etc. gives an accuracy of .15 percent to +15 percent. Using well-defined engineering data and a complete set of requirements gives an accuracy range of .5 percent to +15 percent.
- Eighty to 100 percent of attempts to inherit software not written for inheritance fails [Hihn and Habib-agahi, May 2000, Hihn and Habib-agahi, Sept. 2000].
- An accuracy range of .10 percent to +10 percent requires that 7 percent of a rough order of magnitude budget and schedule be used to develop the plan and budget. Another way to look at this is to consider the percentage of total job calendar time required. When using existing technology, 8 percent of calendar/budget should be allocated to plan development. When high technology is used, then 18 percent of calendar/budget should be allocated to plan development [Remer, 1998].
- According to Boehm [Boehm, et. al., 2000], the impacts of certain risk drivers can be significantly higher than the JPL study:
  - Requirements volatility can increase cost by as much as 62 percent;

- o Concurrent hardware platform development can increase cost by as much as 30 percent;
- o Incorporating anything for the first time, such as new design methods, languages, tools, processes can increase cost by as much as 20 percent, and if there are multiple sources of newness, it can increase cost as much as 100 percent.

6.1.1 Software and Programming Languages

Computers dominate every aspect of modern life. They vary in size and complexity, ranging from mainframe computers used by major companies to personal computers in the home. In addition, microcomputers are used in consumer goods, such as automobile engines, televisions, and microwave ovens. Computers operate based on sets of instructions contained in programs. Computer software may be defined as "computer programs, procedures, rules, and associated documentation and data, pertaining to the operation of a computer system" (IEEE, 1983).

There is a growing trend for programs to use languages that more closely resemble the spoken language. For example, programs for spreadsheets, word processors, and similar applications are often written in a Very Higher-Order Language (VHOL). The advantages of VHOLs are that they allow a person with little or no programming background to interact with a computer. Examples of VHOLs are SQL, Excel, Smalltalk, HTML, and Mathcad. Like size, programming languages have a significant effect on overall software costs.

6.1.2 Software Development Methodologies

Regardless of how software is programmed, its development follows certain steps or phases, and it must be supported (i.e., maintained) after that. The combination of software development and support activities is referred to as the software life cycle. Software development processes describe the methodologies and tools used by an organization, and are key drivers in determining estimated software costs.

There are a number of software development techniques organizations can use, each having a different effect on software costs. One generic software process, used as a framework for many systems currently being developed or supported, is IEEE/EIA Standard 12207, *Standard for Information Technology - Software Life Cycle Processes*, or relevant International Organization for Standardization (ISO) standards (IEEE, 1998). IEEE/EIA Standard 12207 defines a set of recommended development activities and documentation alternatives for software intensive systems. This standard is compatible with a number of different software development methods, including the waterfall model. Figure 6.2 shows the software life cycle phases associated with the waterfall.

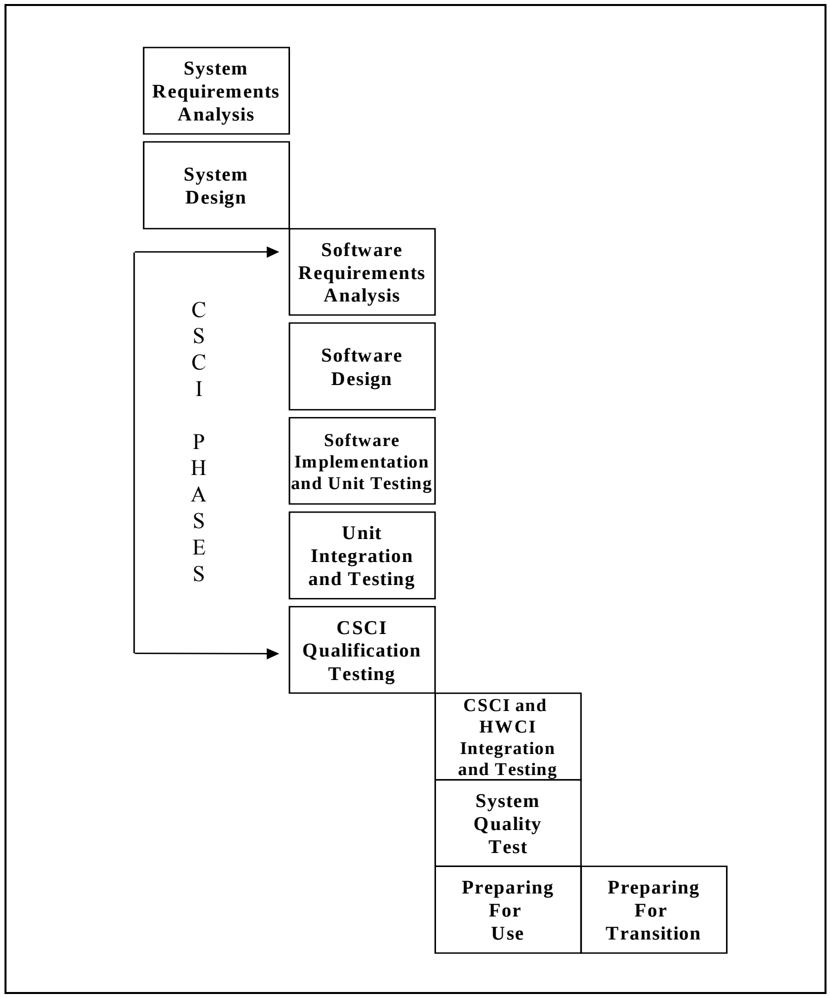


Figure 6.2 Software Waterfall Model Development Phases

Before discussing the development process, it is important to understand the software hierarchy. Figure 6.3 illustrates the software hierarchy commonly used for complex DoD software and MIS systems. Generally, a system (e.g., F-22 fighter aircraft) is partitioned into various subsystems (e.g., avionics) and, at times, prime and critical items (e.g., attack radar). These subsystems or items are further partitioned into computer software configuration items (SCIs) and hardware configuration items (HWCIs). An SCI is defined as an aggregation of software that satisfies a common end-use function. When SCIs are large (e.g., exceed 100,000 LOCs), they are again partitioned into more manageable tiers, called software units (SUs). The lowest-level SUs generally contain between 100 and 200 LOCs. The structure and number of SU tiers depends on the nature and complexity of the particular SCI.

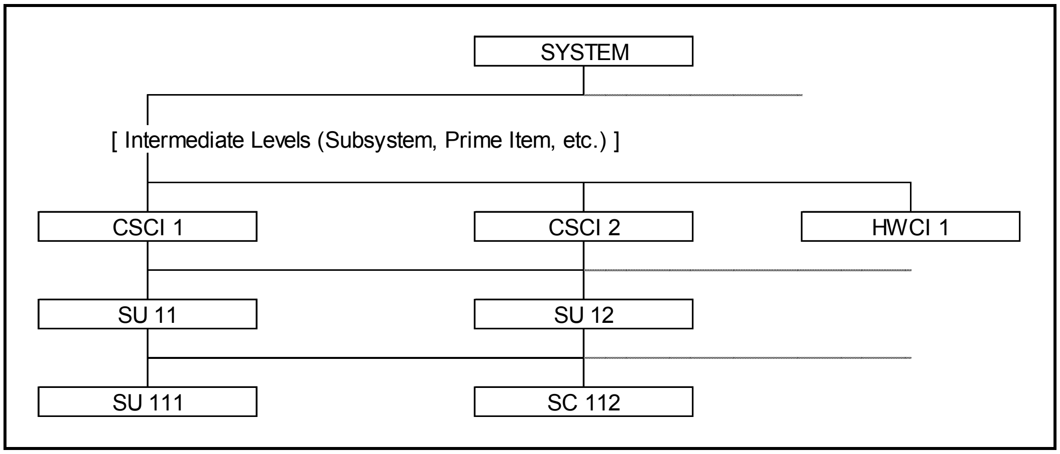


Figure 6.3 Software Hierarchy

The structure in Figure 6.3 is an example of a product-oriented work breakdown structure (WBS), which is a management technique used to subdivide a system into its components. WBSs are generally product-oriented family trees composed of hardware, software, services, and other work tasks. A WBS defines the product(s) to be developed, and relates the work elements to each other and the end product. DoD Handbook *Work Breakdown Structures for Defense Materiel Items* (MIL-HDBK-881A) discusses WBSs in detail.

During the first two phases (system requirements analysis and system design) of the software development process (Figure 6.2), the system level requirements are partitioned into SCI and HWCI level requirements (Figure 6.3). Each SCI is then developed using a SCI life cycle process similar to that shown in Figure 6.2.

During the software requirements analysis phase, specific SCI requirements are defined in detail. During the software design phase, software requirements are refined to the SU level and partitioned into modules where functions, inputs, outputs, and constraints are defined. Generally, once software is completely designed it can be coded (i.e., programmed).

The last three SCI level phases involve: writing source code (e.g., C++ language statements) for each SU; testing each SU; integrating and testing aggregates of SUs; and performing qualification testing on the overall SCI to ensure all requirements are successfully met. After individual SCIs are tested, aggregates of HWCIs and SCIs are integrated and tested. Next, qualification testing is performed on the entire system to ensure the system-level requirements are met. After testing is complete, the software is transferred to the using and supporting agencies.

During each software development phase, a number of other key activities may occur, such as: software project management, software configuration management, and software quality assurance. Each of the activities performed for each discipline, during each phase, can be organized into an activity WBS for each. This WBS can be used with the product WBS as a basis for management reporting and tracking for the SCI.

The software life cycle (i.e., SCI) phases shown in Figure 6.2 do not have to occur sequentially, as the illustration may imply. Many modern development practices can result in a different order of activities or even in a combination (or overlapping) of activities as explained in the following discussions of alternative software development methodologies. The type of methodology used generally has a significant impact on development, maintenance, and total life cycle costs.

6.1.2.1 Waterfall or "Grand Design"

The waterfall methodology is also referred to as the "traditional" software development method. The waterfall emphasizes up-front requirements and design activities and typically requires significant documentation (e.g., specifications, user manuals). The waterfall method was developed in 1970 by W. W. Royce to establish a disciplined approach for software development (Boehm, 1981). It was considered a superior method to the "code-and-fix" practices previously used.

The waterfall approach has certain limitations, including:

- No working product is produced until the last activity is finished (i.e., testing in Figure 6.2);
- Products of preceding SCI phases are usually documents, which tend to be lengthy and cumbersome;
- When problems arise early in the program (e.g., misstated SCI requirements), they may not be discovered until the final product is delivered (at this point it would be expensive and time consuming to correct these problems).

Although this method is still widely used, most software experts recommend that it be used with caution.

6.1.2.2 Evolutionary Development

This methodology involves the initial development of an operational product, and then the continual creation of more refined versions (i.e., iterations) of it. Successive iterations generally follow the SCI activities highlighted in Figure 6.2. During the first iteration, core capabilities are developed and fielded. The software is developed with a modular design so additional capabilities and refinements can be added by the iterations. The advantage of this method is that a working product is available for users early in the development process, which helps them assess the product and provide inputs for the enhanced iterations. One drawback, though, is that the final version can require more time and effort than would be expended under the waterfall method.

6.1.2.3 Incremental Development

The incremental development methodology builds a software product through a series of increments of increasing functional capability, and is characterized by a build-a-little, test-a-little approach. It provides users early participation in a product's development, and can result in savings through the testing of smaller increments. As with evolutionary development, users design later versions by working with earlier ones. The method is not suitable for all programs because partitioning the software into suitable increments is difficult.

6.1.2.4 Prototyping

Prototyping involves the development of an experimental product that demonstrates software requirements for the end users, who get a better understanding of these requirements by working with the prototype. Computer-aided software engineering (CASE) tools facilitate the use of this methodology. While prototyping improves requirements definition, the prototype must not be taken as a "final" product, because this could increase long-term support costs.

6.1.2.5 Spiral Development

This approach views software development as a spiral, with radial distance as a measure of cost or effort, and angular displacement as a measure of progress. One cycle of the spiral usually represents a development phase, such as requirements analysis or design. During each cycle, objectives are formulated, alternative analysis performed, risk analysis conducted, and one or more products delivered. The advantages of the spiral model are that it emphasizes evaluation of alternatives using risk analysis, and provides flexibility to the software development process by combining basic waterfall building blocks with evolutionary or incremental prototyping approaches.

6.1.2.6 Object-Oriented Development

This methodology differs from traditional development in that procedures and data are combined into unified objects. A system is viewed as a collection of classes and objects, and their associated relationships. This is not a separate development method per se, and can be used with other methods (e.g., waterfall, evolutionary, incremental). It can also facilitate software reusability and supportability. Appendix D lists several societies that can provide additional information.

6.1.3 Software Support

Software must be maintained, or supported, after it is developed. Software maintenance includes such activities as adding more capabilities, deleting obsolete capabilities, modifying software to address a change in the environment or to better interface with the host computer, and performing activities necessary to keep software operational. Software support can also be called "software redevelopment" since its tasks repeat all, or some, of the software development phases.

Figure 6.4 explains support categories and gives the relative percentage of effort generally expended on each one. Note that corrective support activities, which many people regard as the sole software maintenance activity, generally account for only 17 percent of the total support effort.

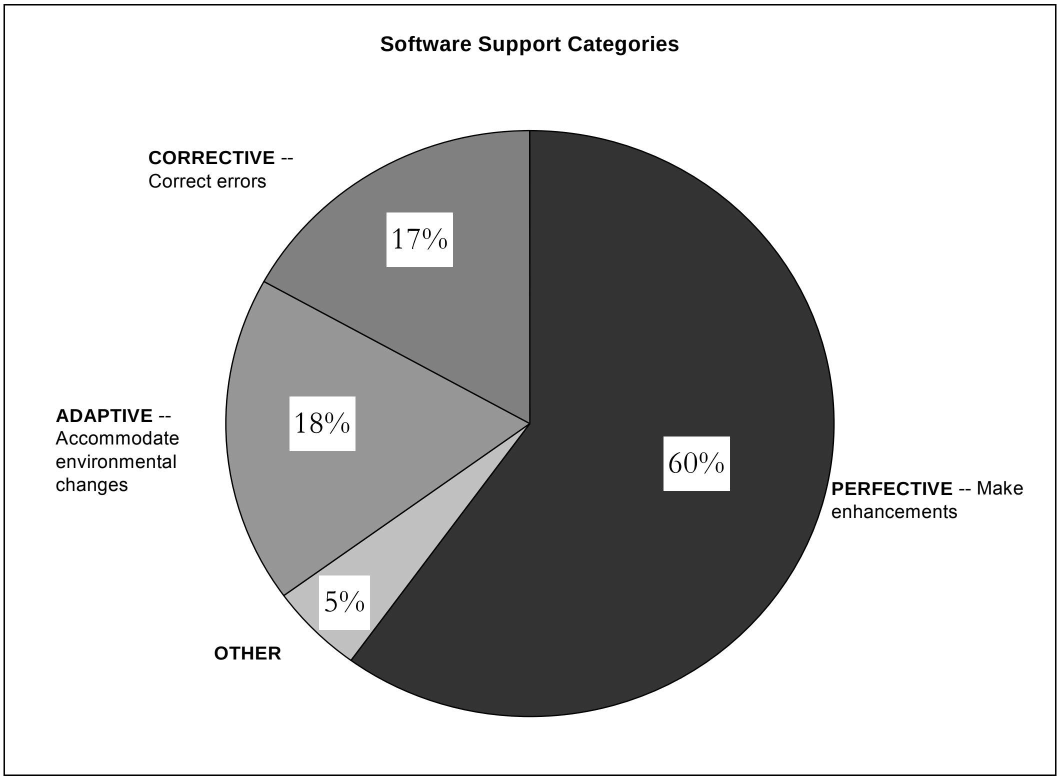


Figure 6.4 Software Support Categories

Software support is expensive, and can exceed the total cost of development. Unfortunately, the techniques often used to estimate support costs are ad-hoc; software support costs are often funded through "level-of-effort" (LOE) type contracts, and are not based on specific support requirements.

6.1.4 Software Capability Maturity Model Integration (CMMI) Models

6.1.4.1 Background

In 1987, the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU), developed a methodology for assessing organizations' software capabilities (Paulk, 1993). This became the framework for the Software Capability Maturity Model (CMM). The CMM was initially developed for the Government to evaluate an organization's ability to perform software development and maintenance work on Government contracts. The CMM was a mainstay for both government and industry during the 1990s; however, SEI decided to replace the CMM in 2001 with a suite of CMM Integration (CMMI) models.

According to CMU, CMMI best practices improve upon the CMM by enabling an organization to:

- More explicitly link management and engineering activities to business objectives;
- Expand the scope of and visibility into the product life cycle and engineering activities to ensure that the product or service meets customer expectations;
- Incorporate lessons learned from additional areas of best practice (e.g., measurement, risk management, and supplier management);
- Implement more robust high-maturity practices;
- Address additional organizational functions critical to its products and services;
- More fully comply with relevant standards (SEI, 2001).

There are actually four CMMI models, with two versions of each: continuous and staged. The staged version of the CMMI for systems and software engineering (CMMI-SE/SW) is discussed here since it tracks most closely with the CMM.

The CMMI-SE/SW has five levels of software process maturity. Figure 6.5 shows the characteristics associated with each level. These characteristics are typically demonstrated by organizations at that level. The levels are sometimes used as key parameters (i.e., inputs) by complex parametric models, and the characteristics may be used to indicate process improvements that need to be implemented before an organization can advance to the next level of maturity.

Maturity Level	Description
Level 1 **Initial**	The software process is characterized as ad-hoc, and occasionally chaotic. Project schedules, budgets, functionality, and quality are generally unpredictable. The organization may succeed, but frequently overruns budgets and schedules.
Level 2 **Managed**	The organization insures that requirements are managed and that processes are planned, performed, measured, and controlled. Standards and processes are documented, but may vary from project to project.

Maturity Level	Description
Level 3 **Defined**	The organization's software processes for both management and engineering activities are well characterized and understood, and are described in detail in standards, procedures, tools, and methods. All projects use an approved, tailored version of the organization's standard software process for developing and maintaining software.
Level 4 **Quantitatively Managed**	The organization controls and measures their performance and the quality of projects using statistical and other quantitative techniques. Detailed measures of the software process and product quality are collected and statistically analyzed.
Level 5 **Optimizing**	The organization focuses on continually improving process performance through innovative and incremental technological improvements. Improvements are based on a quantitative understanding of the causes of variation inherent in processes.

Figure 6.5 Staged CMMI-SE/SW Maturity Level Descriptions

6.1.4.2 Process Areas

For each staged CMMI-SE/SW maturity level (except Level 1), an organization must achieve a number of specific goals and practices for certain process areas. Figure 6.6 lists the required process areas by maturity level. An organization is expected to successfully perform all process areas at each level (and all lower levels) to attain that maturity level; however, tailoring is allowed in special circumstances.

Maturity Level	Process Areas
Level 1 **Initial**	None Required
Level 2 **Managed**	Requirements Management Project Planning Product Monitoring and Control Supplier Agreement Management Measurement and Analysis Product and Process Quality Assurance Configuration Management

Maturity Level	Process Areas
Level 3 **Defined**	Decision Analysis and Resolution Integrated Supplier Management Integrated Teaming Integrated Project Management Organizational Environment for Integration Organizational Process Focus Organizational Training Organization Process Definition Product Integration Requirements Development Risk Management Technical Solution Validation Verification
Level 4 **Quantitatively Managed**	Organizational Process Performance Quantitative Project Management
Level 5 **Optimizing**	Causal Analysis and Resolution Organizational Innovation and Development

Figure 6.6 Process Areas for Staged CMMI-SE/SW Maturity Levels

Many of the process areas focus on an organization's strengths and weaknesses concerning certain topics (e.g., software development methodology). A detailed explanation of the staged CMMI-SE/SW (including all process areas) is beyond the scope of this Handbook; however, three process areas associated with Level 2 (Managed) have specific goals and practices that deal with software estimating: project planning, measurement and analysis, and product monitoring and control. The specific goals and practices for these process areas are described, below.

The purpose of software project planning is to establish and maintain estimates of project planning parameters. Specific practices for doing this include:

- Establishing a top-level work breakdown structure to estimate the scope of the project;
- Establishing and maintaining estimates of the work products and tasks;
- Defining the project life cycle phases used to scope the planning effort;
- Establishing the schedule and cost for work tasks based on estimation rationale.

Periodic project measurement and analysis as well as monitoring and control tracks the performance and progress of a project and compares the performance to the project baseline plan to identify trends and problem areas. Specific practices include:

- Periodically reviewing the project's progress, performance, and issues;
- Reviewing the accomplishments and results of the project at selected project milestones;
- Monitoring actual and earned values of project planning parameters against the project baseline plan;
- Monitoring commitments against those identified in the project baseline or estimate to complete plan;
- Monitoring risks against those identified in the project plan;
- Monitoring the management of the project to the project's baseline plan;
- Monitoring stakeholder involvement in the project plan.

6.1.4.3 Additional Comments on CMM and CMMI

Organizations that have implemented software process improvements resulting from CMM and CMMI evaluations have generally achieved many benefits, including significant cost savings and significant returns-on-investment. In addition, many Government buying activities want contractors to be certified at a particular level before considering them for contract award. Because the CMM and, now, the CMMI are often used as a basis for source selection, organizations have committed substantial resources to implement software process improvements. Software estimates should incorporate the benefits resulting from CMMI related software process improvements, as well as benefits derived from other technologies, such as integration of commercial off-the-shelf (COTS) components and reuse.

The CMMI models are one of many software process improvement methodologies the SEI has developed in recent years.

6.1.5 Manager's Checklist for Validating Software Cost and Schedule Estimates

The SEI developed a CMM checklist that helps managers assess the credibility of software cost and schedule estimates (Park, 1995). It provides issues to address and questions to ask when determining whether or not to use a software estimate. Each question deals with evidence that, if present, supports the credibility of the estimate.

6.1.6 Software Estimating Techniques

Understanding software parametric estimating requires knowledge about basic software estimating methods. Boehm (1981) discusses different types of models and methods for cost estimation, including algorithmic, expert judgment, analogy, Parkinson, price-to-win, bottoms-up, and top-down.

Figure 6.7 summarizes the features, advantages, and disadvantages of four of these. The parametric model category is then discussed.

Model Category	Description	Advantages	Limitations
Analogy	Compare project with past similar projects	Estimates are based on actual experience	Truly similar projects must exist
Expert Judgment	Consult with one or more experts	Little or no historical data is needed; good for new or unique projects	Experts tend to be biased; knowledge level is sometimes questionable
Bottoms-Up	Individuals assess each component and then component estimates are summed to calculate the total estimate	Accurate estimates are possible because of detailed basis of estimate (BOE); promotes individual responsibility	Methods are time-consuming; detailed data may not be available, especially early in a program; integration costs are sometimes disregarded
Parametric Models	Perform overall estimate using design parameters and mathematical algorithms	Models are usually fast and easy to use, and useful early in a program; they are also objective and repeatable	Models can be inaccurate if not properly calibrated and validated; it is possible that historical data used for calibration may not be relevant to new programs

Figure 6.7 Categories of Software Cost Models

Parametric models generate estimates using statistical relationships, and relate dependent variables (i.e., cost and/or schedule) to one or more independent variables (i.e., parameters). Parametric estimating techniques for software projects generally estimate overall system or SCI costs based on a software program's design characteristics. These overall costs can be partitioned among the lower-level SUs or life cycle phases. The advantages of parametric models are that they are fast and easy to use, require little detailed information (after they are calibrated), and capture total system or SCI-level costs (including costs for integration activities). In addition, parametric estimating techniques can be as (if not more) accurate as other estimating techniques, if they are properly calibrated and validated. Because of these advantages, parametric models are generally DoD's software estimating technique of choice. Section 6.2 looks at several complex parametric models that both Industry and Government use.

6.2 Overview of Software Parametric Cost Models

Many sophisticated parametric software estimating models use multiple parameters to compute software costs and effort. This section discusses common models and provides a basic understanding of some of their common features. The discussion focuses on background, principal inputs (i.e., parameters), processing, and principal outputs, as well as cost estimating capabilities for software support, because of its growing importance. Appendix A contains contact information for the model vendors.

Many software parametric cost models depend upon a measured input for source lines of code (SLOC) as the primary input for the cost-driving variable for mass (equivalent to weight in a hardware model). Over the years, SLOC has been the standard cost-driver for parametric software models. The definitions for SLOC can vary depending upon the parametric model, so the user must ensure that he or she is following the definition appropriate to the model being used. There are other definitions for mass that can be employed. Some of these are described below. Selection of the appropriate cost driver is of critical importance.

6.2.1 Function Point Models

Function points are weighted sums of five factors related to user requirements: inputs, outputs, logic files, inquiries, and interfaces. They are parametric models because they use design parameters to estimate size. However, these parameters were not developed using regression analysis procedures; instead, they use the five program factors (inputs, outputs, logic files, and so forth) to estimate software size.

Function point analyses have been performed on more than 30 data processing programs. The resultant conclusions were that function points are not only a valid predictor of software size, but are also superior to SLOC as a predictor of software development cost or effort. Most models can use function points as an alternative to SLOC for estimating software size.

Figure 6.8 shows how traditional function points (sometimes called Albrecht function points) are computed. The user must determine the number of external inputs (EI), external outputs (EO), external inquiries (EQ), internal files (ILF), and external interfaces (EIF) in the program. This determines a measure called "basic" function points. The user can then refine this measure by considering the complexity level of each function point and the 14 complexity adjustment factors related to the overall program (see Figure 6.8). The attributes used for function points are:

- **External inputs (EI).** All unique data or control inputs that cross the system boundary and cause processing to occur (e.g., input screens and tables).

- **External outputs (EO).** All unique data or control outputs that cross the system boundary after processing has occurred (e.g., output screens and reports).
- **External inquiries (EQ).** All unique transactions that cross the system boundary to make active demands on the system (e.g., prompts and interrupts).
- **Internal files (ILF).** All logical data groupings that are stored within a system according to some pre-defined conceptual schema (e.g., databases and directories).
- **External interfaces (EIF).** All unique files or programs that cross the system boundary and are shared with at least one other system or application (e.g., shared databases and shared mathematical routines).

- "Basic" Function Points (BFP): 4(EI) + 5(EO) + 4(EQ) + 10(ILF) + 7(EIF) (with ±25% Complexity Adjustment)
- Unadjusted Function Points (UFP): Weight Five Attributes as Simple, Average, or Complex

Attribute	Complexity			Total
	Simple	Average	Complex	
EI	3	4	6	
EO	4	5	7	
EQ	3	4	6 (or 7)	
ILF	7	10	15	
EIF	5	7	10	

- Adjusted Function Points (AFP): UFP (0.65 + [0.01(CA)])
  (CA is Complexity Adjustment: Sum of 14 Factors, Rated 1 to 5 for Influence [0 - None, 1 - Little, 2 - Moderate, 3 - Average, 4 - Significant, 5 - Strong]; Ratings Defined for Each Factor)

14 Factors:	
1. Data Communications	2. Distributed Data Processing
3. Performance Objectives	4. Heavily-Used Configuration
5. Transaction Rate	6. On-Line Data Entry
7. End-user Efficiency	8. On-Line Update
9. Complex Processing	10. Reusability
11. Conversion and Installation Ease	12. Operational Ease
13. Multiple Site Usage	14. Facilitate Change

Figure 6.8 Traditional Function Point Computations

The excellent results obtained from Albrecht and Gaffney's research are a noted strength of function-point models. In addition, the International Function Points User's Group (IFPUG), which meets twice a year, and periodically publishes a guide to counting and using function points (Garmus, 2001), performs ongoing research. Proponents of function point size estimation state that function point counts can be made early in a program, during requirements analysis or preliminary design. Another strength, according to Capers Jones (Jones, 1995), is that they provide a more realistic measure of productivity because SLOC-per-person-per-month measures tend to penalize HOLs (e.g., ADA, C++).

However, function points do have disadvantages, since they are often harder to visualize (i.e., functions points are concepts), where SLOCs can be seen (e.g., on a code listing). Function points are also less widely used than SLOC for most applications, and have only been studied extensively for business or data processing applications, though attempts to adapt the function point concept to real-time and scientific environments have been made.

6.2.2 Conversion of Function Points to SLOC

It is sometimes necessary to convert from SLOC to function points, or vice-versa. Several software cost models, such as The Early Design Model of COCOMO II, allow the user to input function points (or a variant of a function point), though they must convert function points to SLOC (because the model's algorithms are based on SLOC). The opposite situation can occur in other models where SLOC inputs must be converted to function points. This conversion process is sometimes called "backfiring."

To help in this conversion process, sets of SLOC to function point ratios have been developed. See Figure 6.9.

Language levels are useful for converting size from one language to another, and for assessing relative language productivity (although the relationship between language level and productivity is not linear).

Language	Jones	Jones	Galorath	Reifer
	Language Level	SLOC/FP	SLOC/FP	SLOC/FP
Assembler	1	320	320	400
COBOL	3	107	61	100
FORTRAN	3	107	58	105
ADA (1983)	4.5	71	71	72
PROLOG	5.0	64	61	64
Pascal	3.5	91	71	70
PL/1	4.0	80	71	65

Figure 6.9 SLOC-Per-Function Point Ratios

While function point to SLOC conversion ratios are useful, and often sometimes necessary, they should be used with caution.

Figure 6.9 illustrates that, while researchers may agree on the ratios for some languages such as ADA, they differ on the ratios for others, such as Pascal and PL/1. Furthermore, there was considerable variance for these ratios within the databases. Therefore, for some languages it appears that backfiring should not be used, and for cost estimation it is probably best to use a model for which the algorithms are based on the user's size measure (i.e., calibrated parametric sizing models).

6.2.3 Object Points

Other sizing methods were developed to address modern programming applications. Currently, object points are used in development environments using integrated CASE tools (although they may have other applications). CASE tools automate the processes associated with software development and support activities and, when used correctly, can have a significant impact on productivity levels as well as quality factors associated with software costs, such as rework.

The four object types used include:

- **Rule Sets.** A collection of instructions and routines written with a CASE tool's high-level language (these are analogous to "programs" when third-generation languages (3GL) such as FORTRAN or COBOL are used).
- **Third Generation Language Modules.** Existing procedures written in a 3GL.
- **Screen Definitions.** Logical representations of on-screen images.
- **User Reports.** Specific types of reports.

Two object-based measures are obtained from these object types. The first, object counts, is merely a sum of the number of instances of each object type and is analogous to basic function points. Object points are a sum of object instances for each type, times an effort weight for each type.

The average effort weight for each type is as follows:

- Rule Sets: 3 Days;
- 3GL Modules: 10 Days;
- Screen Definitions: 2 Days;
- User Reports: 5 Days.

Therefore, object points are an estimation of effort needed for an integrated CASE tool development environment. Application points, a variant of object points, are currently used in the COCOMO II Application Composition model,

6.2.4 Use Case Points (UCPs)

With the increasing popularity of Unified Modeling Language (UML) and similar programming languages, UCPs as a measure of software size are receiving increased attention. UCPs are a sum of actors and use cases, each adjusted for complexity. Use cases are further adjusted by technical complexity from 13 technical factors, and environmental complexity from 8 environmental factors. The result is a count of adjusted UCPs. UCPs can then be used to estimate size for another measure, such as SLOC. If productivity rates are known (UCP/PM), UCPs can be used to directly estimate effort.

6.2.5 Normalized Use Cases (NUCs)

Initial work in UCPs advanced and diverged in 2002 from other use case work that a new name was coined, Normalized Use Cases, or NUCs. NUCs include actors, number of use cases, complexity factors, and other analyses. These have been applied to several systems and have been able to provide up-front size estimation. These were completed systems with blind application of the sizing model. Additional research is currently being performed.

6.3 Cost Model Selection

With the multitude of software cost and sizing models available, selecting the appropriate tool can be difficult. A four-step approach can be used in the selection process:

1. Determine user needs;
2. Select candidate models;
3. Choose the most appropriate model or models;
4. Reevaluate the choice.

6.3.1 Step 1: Determine User Needs

This first step is the most crucial. Different models are best for different applications, and the user should understand the unique requirements of the program. The user should first write a general statement of the organization's needs, then expand the information in more detail. A "weighted factors approach," such as that illustrated in Figure 6.10, can clearly define each unique situation. The list of factors and weightings shown reflects their importance to the user’s organization. (They are presented as an example, and may be quite different for other organizations.) Also, the listing of factors and assignment of weightings can be subjective. However, this approach provides a framework for considering qualitative evaluation factors.

Factor	Importance Rating	Model Ratings			Sub-Factor Products		
		A	B	C	A	B	C
Input Data Availability	10	10	9	7	100	90	70
Design Evaluation Criteria	9	10	6	7	90	54	63
Ease of Use	8	8	9	6	64	72	48
Ease of Calibration	6	2	5	5	12	30	30
Database Validity	5	7	7	4	35	35	20
Currentness	5	3	5	5	15	25	25
Accessibility	4	6	9	4	24	36	16
Range of Applicability	2	1	7	10	2	14	20
Ease of Modification	1	3	4	2	3	4	2
Weighted Totals					**345**	**360**	**294**

Figure 6.10 The Weighted Factors Approach

6.3.2 Step 2: Select Candidate Models

The second step is to select a set of candidate models that meet the needs identified through Step 1. An examination of needs can point out the most suitable models. For size estimation, various categories of models (e.g., analogy, bottom-up, expert judgment, or parametric) can be selected. However, for cost models, the choices will probably focus on parametric models. Once the category or categories are identified, candidate models can be selected.

6.3.3 Step 3: Choose the Most Appropriate Model or Models

The user should perform both qualitative and quantitative (accuracy) assessments of the candidate models selected in Step 2, and choose the best model or models for their organization. For software estimates, it is recommended that two models be selected for routine use: one as the primary model and one for crosschecking its results. A study by Coggins and Russell (Coggins, 1993) showed that software cost models, even given "equivalent" inputs, produce significantly different cost and schedule estimates. They concluded that a user should learn one or two models well, instead of trying to use several different models. Nevertheless, other models can still be used, even if only for future consideration (discussed in Step 4).

Users must become familiar with each of the candidates to choose the most effective model. This often involves attending a training course and using the model for several months. Once the user becomes sufficiently familiar with the models, the selection process can begin. It is highly desirable that users perform their own studies, and not rely solely on outside information. Nevertheless, validation studies performed by outside agencies can certainly help the user in the model selection process. An excellent example is a study by the Institute for Defense Analysis (Bailey, 1986), which compared and evaluated features of most of the cost models then in use by Industry and Government. While outside studies can provide valuable information, they should be used as supplementary material since they may not reflect the unique features of the user's environment.

The Weighted-Factors Approach (Figure 6.10) can help with the qualitative assessment of candidate models. The user assigns a weight to each factor (Step 1), assigns a rating between "1" and "10" to each model (based on how well it addresses each factor), multiplies the model and importance ratings, and sums the results. The highest total can indicate the best model alternative (e.g., Model B in Figure 6.10). However, models that are close in score to the highest (e.g., Model A in Figure 6.10) should also be examined. Since there is some subjectivity in this process, small differences may be negligible. Again, while the Weighted-Factors Approach is somewhat subjective, it can help a user consider what is important in model selection and in quantifying the rating process.

For quantitative assessments, or in determining whether the models meet accuracy requirements, users should calibrate the models, then run them against projects for which the user has historical data that was not used during the calibration process. This approach is often arduous, but essential if a user truly wants to identify the model that is most suitable (i.e., most accurate) for the application.

6.3.4 Step 4: Reevaluate the Choice

User needs and models can change over time. Many commercial models are updated every year, and major refinements occur every few years. New models occasionally appear that could be more suitable than the current models. Therefore, users should reevaluate their selections every few years. There is no reason to be "married" to a particular model, or models, for life unless they continue to be the best available.

The four-step approach can help a user in model selection. The most crucial step in this process is the first: user needs determination. The remaining steps hinge on its success. The four-step approach is sometimes laborious, but the benefits of improved estimating make it worthwhile.

6.4 Intelligent Use of Software Models

A bounty of software cost and size models are available, many of them very sophisticated, which could lull an analyst or manager into an over-dependence on them. Some managers believe that "models, not estimators, are responsible for estimates" (Park, 1989). Software models however, are not magic boxes; they are only as good as the input data used. Even models have limitations. For example, parametric models can be inaccurate if they have not been adequately calibrated and validated. Furthermore, models are not always useful in analyzing non-cost factors that may impact certain decision-making. Therefore, a manager must be able to recognize the capabilities and limitations of models and use them intelligently. An example is provided at the end of this section.

6.4.1 Input Data

One problem with parametric models is that their effort and schedule estimates may be very sensitive to changes in input parameters. For example, in most cost models, changes in program size result in at least an equivalent percentage change in cost or effort. Other input changes can have dramatic effects; for instance, changing the two COCOMO II personnel capability ratings, (analyst capability (ACAP) and programmer capability (PCAP), from "very high" to "very low" will result in a 350 percent increase in effort required. All models have one or more inputs for which small changes result in large changes in effort and, perhaps, schedule.

The input data problem is compounded by the fact that some inputs are difficult to obtain, especially early in a program (e.g., software size). Other inputs are subjective and often difficult to determine; personnel parameter data are especially difficult to collect. Even "objective" inputs like security requirements may be difficult to confirm early in a program, and later changes may result in substantially different cost and schedule estimates. Some sensitive inputs such as productivity factors should be calibrated from past data. If data are not available, or if consistent values of these parameters cannot be calibrated, the model's usefulness may be questionable.

A manager or analyst must spend considerable time and effort to obtain quality input information. Ideally, a team of personnel knowledgeable in both software estimating and technical issues should perform a software cost estimate. A software cost analyst must work with engineering or technical personnel to determine some of the "hard" inputs such as size and complexity. The analyst should also try to determine "soft" inputs (e.g., analysts' capability—ACAP) by working with appropriate personnel in the organization and, if necessary, performing a Delphi survey or similar expert judgment technique. Finally, an analyst or team should calibrate the models to the particular environment, a time-consuming but worthwhile exercise. As previously discussed, model calibration should improve model accuracy.

6.4.2 Model Validation

If a model will be used to develop estimates for proposals that will be submitted to the Government or a higher tier contractor, its accuracy should be addressed through the validation process. Validation is defined as the process, or act, of demonstrating a calibrated model's ability to function as a credible forward-estimating tool. A parametric model, such as one for software estimating, should be implemented as part of a contractor's estimating system. For a model to be considered an acceptable (or valid) estimating technique, an organization should be able to demonstrate that:

- Key personnel are experienced and have received adequate training;
- Complex model calibrations were performed and documented;
- Estimating procedures are established to enforce consistency in the calibration process as well as in application on proposals;
- Parametric techniques are good predictors of future costs.

Chapter 7, Government Compliance, discuss the criteria for adequate parametric estimating systems.

6.4.3 COCOMO II

Because it is an open book model, COCOMO II (REVIC is also) is an example software estimating tool that can be used for performing model-based estimates. USC COCOMO II is a tool developed by the Center for Software Engineering (CSE) at the University of Southern California (USC), headed by Dr. Barry Boehm.

Unlike most other cost estimation models, COCOMO II (www.sunset.usc.edu) is an open model, so all of the details are published. There are different versions of the model, one for early software design phases (the Early Design Model) and one for later software development phases (the Post-Architecture Model). The amount of information available during the different phases of software development varies, and COCOMO II incorporates this by requiring fewer cost drivers during the early design phase of development versus the post-architecture phases. This tool allows for estimation by modules and distinguishes between new development and reused/adapted software.

6.4.4 Example: REVIC – Revised COCOMO Model

The REVIC equations are:

1. MM= A (KDSI)↑B x Fi
2. TDEV = C(MM)↑D

Equation (1) predicts the manpower in man months (MM) based on the estimated lines of code to be developed (KDSI = Delivered Source Instructions in thousands) and the product of a group of Environmental factors (Fi). The coefficients (A, C), exponents (B, D) and the factors (Fi) are determined by statistical analysis from a database of completed projects. These variables attempt to account for the variations in the total development environment (such as programmer's capabilities or experience with the hardware or software) tending to increase or decrease the total effort and schedule.

The results from equation (1) are input to equation (2) to determine the resulting schedule (TDEV = Development Time) in months needed to perform the complete development. COCOMO provides a set of tables distributing the effort and schedule to the phases of development (system engineering, preliminary design, critical design, and so forth) and activities (system analysis, coding, test planning, and so forth) as a percentage of the total effort.

6.4.5 Example: Software Experience

Suppose that the following formula, based upon normalized data, organizational calibration and validation, has been derived to reflect an organization’s software experience (the formula comes from the REVIC Model):

$$MM = k\,(KSLOC)\uparrow b \times EAF$$

Where:

MM	=	Man Months of effort
k	=	Value from lookup table (from organization, operational environment, programming language, and so forth).
b	=	Exponent value from lookup table
KSLOC	=	Thousands of source lines of code
EAF	=	Effort Adjustment Factor (from operational environment)

If $MM = k(KSLOC)\uparrow b \times EAF$

Then:

MM	=	Man Months
k	=	3.22
KSLOC	=	36.300
B	=	1.2
EAF	=	1.0
MM	=	$3.22\ (36.3)\uparrow 1.2 \times 1.0$
MM	=	240

6.5 Future Directions of Software Estimating

If the current software estimating environment appears challenging, the future will certainly be more so. Advances in languages, development methodologies, and other areas will have to be addressed by future software estimating models and methodologies. Some of the current and future challenges to software estimating are highlighted.

6.5.1 New Development and Support Concepts

As software development technology matures, changes in many development and support concepts will occur that may impact software estimates. There will probably be an increased use of some of the leading edge development methods, such as object-oriented development. These changes would affect labor resource loading assumptions, such as those used in various cost models. As these and other new development methods become more popular, cost models will require enhancements to better address their cost impacts. An analyst estimating a software program that uses modern development methodologies should investigate the model's user's manual or contact the model vendor for additional information.

6.5.2 Reuse and COTS Integration

An issue of concern to many software managers is the cost of reusing previously developed software for new programs. In addition, a related issue is the cost of integrating COTS programs into new or existing ones. Reusing software can significantly reduce development costs; therefore, COTS programs should not require significant development effort (except for integration, which can be significant). Most software cost models give special consideration to these issues. Most models ask the user for percentages of new design, new code, and retesting required for reused code.

There are several limitations related to COTS. First, there are not large repositories of reusable software components available for general use. Second, the effort may require more than some of the models can provide. Third, software project managers are often reluctant to incorporate reusable software into their programs, because their teams are used to developing programs from scratch. Reuse often involves a change in the organizational infrastructure. Finally, reusable software may be difficult or impossible to support. This can be especially problematic for COTS software if the programs have restricted data rights.

6.5.3 New Cost Models

As software technology matures, models will change or evolve to address these changes. Therefore, the cost analyst or manager can expect to encounter new models or modifications to existing ones.

Model users should ensure they are familiar with the latest editions of these models and obtain retraining as necessary. Some commercial software estimation models as they appear today (2007) are more fully described in Appendix A.

6.6 Lessons Learned

The results of the Parametric Estimating Reinvention Laboratory demonstrated that software parametric models should be implemented as part of an organization's estimating system. Parametric estimating systems should consist of credible databases; adequate policies and procedures containing guidance on data collection and analysis activities, calibration, and validation; and policies and procedures to ensure consistent estimating system operation. Chapter 7, Government Compliance; provides detailed guidance on the Government's expectations related to software estimating using parametric techniques.

The effective implementation of software parametric techniques involves establishing adequate resources to populate software metric databases on a regular basis. Figure 6.11 contains a listing of key metrics that contractors should collect (Grucza, 1997).

Category	Measure
Size (By Language)	• SLOC (New, Reused, Total) / Estimate at Completion • SLOC (New, Reused, Total) / Actuals – Forecast • Document Pages
Effort	• Labor Hours (Direct) • Staff Positions • Cost
Productivity	• LOC/Hour; Pages/Hour
Requirements Stability	• Requirements (Total, Changed)
Schedule	• Requirements Allocated to Components • Requirements Verified • Units Designed • Unites Coded & Units Tested • Units Integrated • Test Cases Executed • Milestone Dates for Software Development Activities • Milestones (Total, Completed)

Category	Measure
Environment	• Throughput • Computer Memory Utilization • Input/Output Channel Utilization
Quality	• Defects (Found, Closed) • Defect Action Item • Peer Reviews
Training	• People, Classes, Hours Taught, Cost • People Taught
Parametric Model Data	• Data Sheets
Risk Management	• Risk Items
Earned Value	• Cost Performance Index or Milestones/Hour
Intergroup Coordination	• All of the Above
Integrated Software Management	• Software Development Plan Updates

Figure 6.11 Key Software Metrics

Parametric Estimating Reinvention Laboratory results demonstrated that integrated product teams (IPTs) are a key practice for implementing new parametric techniques, including those for software estimation (see Appendix J). IPTs should include representatives from a company, key customers, DCMA, and DCAA. Chapter 8, Other Parametric Applications, contains guidance on the IPT processes. IPTs should also jointly participate in training related to the software estimating techniques and/or models being implemented or used.

6.7 Best Practices

During the Parametric Estimating Reinvention Laboratory, an IPT used a complex parametric model for software estimating. At the beginning of its implementation, the contractor IPT members found the most challenging task was obtaining the necessary internal resources (i.e., commitments) to perform data collection and analysis activities. Later the company initiated software process improvement activities consistent with CMMI criteria, for Levels 2 and 3. As previously discussed, this criterion includes establishing databases and metrics for software estimation. The IPT recognized the implementation of a complex parametric software estimating model could be greatly facilitated when done in conjunction with the software process improvement activities related to the CMMI. Of course, if a contractor has already achieved and continues to maintain

Level 2 or 3 status, the implementation of complex parametric models should be greatly enhanced.

6.8 Conclusions

Software model users and managers are continually challenged to make intelligent use of current models and to keep abreast of the impacts of future changes. New languages, new development methods, and new or refined models are a few of the many areas a model user must have current knowledge. Technologies such as graphical user interfaces and artificial intelligence can also affect software estimation. However, current parametric software cost models have many features and capabilities and, when calibrated, can provide detailed and accurate software estimates.

CHAPTER 7
Government Compliance

Despite prevalent misconception, the applicable federal statutes and corresponding regulations need not preclude or limit the use of parametric pricing techniques, including projection of subcontracting costs. As such, contractors are afforded sufficient latitude to implement estimating systems policies and procedures, while still complying with the Truth in Negotiations Act (TINA), Federal Acquisition Regulations (FAR), Defense Federal Acquisition Regulations Supplement (DFARS), and the Cost Accounting Standards (CAS). Accordingly, this chapter discusses the aforementioned procurement laws and regulations as they relate to parametric estimating techniques when cost or pricing data are required to establish fair and reasonable prices, as well as the Government's perspective when auditing and otherwise reviewing parametric proposals and estimating systems.

Additionally, this chapter will:

- Demonstrate how the use of properly calibrated and validated parametric estimating techniques ensures compliance with the Government procurement regulations;
- Delineate not only the key Government procurement regulations in terms of parametric estimating but also identify and discuss estimating system policies and procedures that result in compliance;
- Provide examples of regulatory challenges addressed by the historic Parametric Estimating Reinvention Laboratory integrated product teams (IPTs), as well as other more recent working groups, and describe the processes used to resolve them;
- Delineates characteristics of an acceptable Estimating System, including application to specific proposals, and the corresponding technical and audit emphasis during Government reviews.

7.1 Regulatory Compliance

The proper use of calibrated and validated parametric estimating CERs and parametric models, in tandem with the establishment and consistent adherence to effective estimating policies and procedures, will promote compliance with the applicable procurement statutes and regulations. This section discusses the various regulatory requirements.

7.1.1 The Truth in Negotiations Act (TINA)

TINA (Public Law 10 U.S.C. 2306(a)) requires cost or pricing data be certified as current, complete, and accurate as of the date of contract negotiation (or an otherwise agreed to date) so as to provide a basis for the Government to negotiate reasonable prices. Generally, TINA applies to sole-source procurements in excess of $550,000 (increased to $650,000 effective September 28, 2006), to either prime contracts or subcontracts, awarded after October 11, 2000, unless a FAR 15-403-1(b) exception applies. Exceptions are prices based on adequate price competition, prices of commercial items, prices set by law or regulation, and prices resulting when a contract or subcontract is modified and meets the parameters of FAR 15-403-1(c)(3). Also, in exceptional cases, the head of the contracting activity (HCA) may grant a waiver to the requirement for submitting cost or pricing data.

Cost or pricing data includes all factual data that can be expected reasonably to contribute to the soundness of future cost estimates, as well as to the validity of costs already incurred. For parametric techniques, factual data includes historical data used to calibrate the model (or in building databases for cost estimating relationships (CERs)), such as:

- Technical data (e.g., weights, volume, speed);
- Programmatic data (e.g., project schedules);
- Cost data (e.g., labor hours, overhead rates, G&A expenses);
- Information on management decisions that could have a significant effect on costs (e.g., significant changes to management and manufacturing processes).

Cost or pricing data does not include judgmental data, but does include the factual data on which judgment is based. Like all traditional estimating techniques, parametric estimates contain judgmental elements that are not subject to certification, yet need be disclosed pursuant to Part FAR 15, since they are subject to negotiation.

Specific to parametric techniques, properly calibrated and validated CERs and parametric models, as supported by corresponding company policies and procedures, are expected to be fully compliant with TINA requirements through the cyclical processes of calibration and validation themselves. Accordingly, the matters of currency and completeness of that data should not become issues, provided the frequency of calibration and validation of said data is technically sufficient, and addressed by Government/contractor agreements and approved estimating policies and procedures, as well as their successful implementation.

Additional information relating to the Government expectations when developing CERs is included in the Defense Contract Audit Manual Section 9-1000. In terms of strict interpretation of the law, the key is full disclosure of all factual pricing data, and not whether the said data was necessarily relied upon, particularly for updates and other out-of-cycle data. However, while compliance may be no longer an issue in such instances, the excluded data may be of such significance in determining a fair and reasonable price that its very exclusion may become an issue during the ensuing negotiation.

How to handle out-of-cycle data of significance and how to determine what significance means is perhaps best addressed by agreement so as to avoid such scenarios. In fact, the findings of the Parametric Estimating Reinvention Laboratory support that, in general, the best way for contractors to comply with the TINA requirements is to establish adequate parametric estimating procedures that have been coordinated with their Government representatives. These representatives include the contractor's primary customers, DCMA, and DCAA. The next section discusses some key elements of parametric estimating procedures.

7.1.2 FAR, DFARS, and Estimating Systems

Cost estimating systems are critical to the development of sound and reasonable pricing proposals. DFARS 215.407-5(d), *Estimating Systems*, specifies characteristics of an acceptable estimating system, including the following general criteria:

- Use appropriate source data;
- Application of sound estimating techniques and good judgment;
- Maintenance of a consistent approach;
- Adherence to established estimating policies and procedures.

For parametric pricing in particular, it is crucial that an estimating system policy and procedure be established to address frequency of calibration or the need to incorporate database updates. Calibration, the adjustment of general parameters of a parametric model, should be achieved with sufficient frequency so as to reasonably capture and predict the cost behavior for a particular product or product line at a specific firm and site. Data collection at the specific site is a prerequisite. However, in recognition of the continuous influx of relevant data, the use of cut-off dates in updating the model becomes a consideration. As such, FAR 15.406.2(c) encourages contracting officers and contractors to negotiate an agreement on the criteria used to establish acceptable cut-off dates (e.g., monthly, quarterly, annually).

When used, cut-off dates should be defined for all significant data inputs to the model, and included in a company’s estimating system policies and procedures. Further, contractors should disclose any cut-off dates in their proposal submissions. The parties should revisit the relevancy of the established dates before settling on final pricing agreement, and seek updates in accordance with the contractor’s disclosed procedures. When cut-off dates are not used, companies should have proper procedures to demonstrate that the most current and relevant data were used in developing a parametric based estimate.

Additionally, to ensure data are current, accurate, and complete as of the date of final price agreement, contractors must establish practices for identifying and analyzing any significant data changes so as to determine if out-of-cycle updates are needed. A contractor's estimating policies and procedures should identify the circumstances when an out-of-cycle update is needed.

Examples of some events that may trigger out-of-cycle updates include:

- Implementing new processes significantly impacting costs;
- Completing an additional contract lot reflecting changing costs;
- Reversing make-or-buy decisions;
- Upgrading or otherwise changing specifications, including new materials;
- Implementing major accounting changes, invalidating unadjusted historical data;
- Changing major subcontractors;
- Restructuring/merging.

Overall, the policies and procedures of a contractor's estimating system should include adequate detail to enable the Government to make informed judgments regarding the acceptability of the disclosed parametric estimating practices.

Additionally, the policies and procedures should incorporate:

- Guidelines to determine applicability of parametric estimating techniques;
- Guidelines for collecting and normalizing data, including criteria for determining logical relationships, and their significance;
- Methodologies for calibrating the CERs and parametric models, including identification of the significant cost drivers (e.g., weights, volumes, software lines of code, complexity factors) and the desired input and output parameters;
- Methodologies for validating calibrations to demonstrate accuracy;
- Guidelines for ensuring consistent application of parametric techniques;
- Procedures to ensure all relevant personnel have sufficient training, experience, and guidance to perform parametric estimating tasks in accordance with the disclosed estimating processes;
- Guidelines for performing internal reviews to assess compliance with estimating policies and procedures, including assessing the accuracy of the parametric estimates.

In addition to the establishment of estimating policies and procedures as delineated above, contractors may find it advantageous to execute a memorandum of understanding (MOU) with their cognizant administrative contracting officer (ACO), even though not mandated by regulation.

During the days of the Parametric Estimating Reinvention Laboratory, several contractors used MOUs to further define and refine agreed upon estimating practices so as to preclude the incidence of foreseeable infractions with TINA. The MOUs clarified interpretations of such issues as definition of input data, cut-off dates, procedures for identifying unusual events, and frequency of database updates. The MOUs also established the rules of engagement for building compliant parametric estimating capability, concurrent with development of the associated estimating system policies and procedures.

While such MOUs are generally formalized between a company and its ACO and major customers, DCMA and DCAA are available to provide input and feedback even when a direct signatory party to the agreement. As a best practice, a list of suggested elements to include in a MOU is included as Appendix F**.**

7.1.3 Cost Accounting Standards (CAS)

FAR 30.101, *Cost Accounting Standards*, implements the requirements of Public Law 100-679 (41 U.S.C. 422). As with TINA, properly calibrated and validated parametric estimating techniques should result in compliance with CAS. This is supported by the fact that during the Parametric Estimating Reinvention Laboratory there were no CAS non-compliances noted, even where contractors were instructed to provide estimates at a very high level of detail to afford greater insight into the negotiation process.

In that instance, the buying activity inserted the following provision into a request for proposal (RFP) so as to not prejudice any proposals based upon parametrics.

> "When responding to the cost volume requirements in the RFP, the offeror and their associated subcontractors may submit cost estimates utilizing appropriately validated parametric techniques that are part of their disclosed cost estimating system. These include contemporary cost estimating relationships (CERs), commercially available parametric cost models, and in-house developed parametric cost models. If necessary, reasonable and supportable allocation techniques may be used to spread hours and/or costs to lower levels of the work breakdown structure (WBS). The offeror's use or non-use of the parametric estimating techniques for this proposal will not be a factor (positive or negative) in the evaluation of the offeror's response to the RFP. Cost estimates submitted utilizing such parametric models should produce cost estimates that are reasonable and consistent with its practices used for accumulating and reporting costs, and as such, create a basis for negotiation of price."

The use of CERs and parametric models in themselves do not conflict with estimating to the WBS level any more than use of the more traditional, non-parametric techniques. Additionally, use of parametric methodologies do not automatically create a non-compliance with CAS 401, *Consistency in Estimating,* *Accumulating, and Reporting Costs*, which requires cost accounting practices used to estimate proposal costs be consistent with practices used for accumulating and reporting costs. CAS 401 applies to all estimating techniques, including parametrics.

Case in point, during the Parametric Estimating Reinvention Laboratory, a contractor proposed to estimate costs at a lower level than that used to accumulate and report costs, which would have resulted in a noncompliance with the requirements of CAS 401. The contractor planned, however, to begin accumulating and reporting at the lower level. To preclude the noncompliance, the IPT members therefore established a process the contractor could use in the short-term to estimate at the lower level of detail, while still complying with the requirements of CAS 401. This process required the contractor to provide adequate justification that included:

- Explanation for the need and benefit of estimating the costs at a lower level;
- Reasonable support for the estimated cost;
- A general reconciliation of the lower level detail to the level at which the costs are accumulated and reported.

The only other CAS issue, specifically a CAS 401 issue, that is of concern and merits attention is the additional requirement that the estimating techniques be consistent with the disclosed practices in the CAS Disclosure Statement, not to be confused with disclosure under the Estimating System. In most instances, it is unlikely that the Disclosure Statement would be of such minute detail that inconsistencies would occur. Nonetheless, due diligence is called for.

Likewise, when using complex parametric models, or tasking estimators with developing in-house models that are unfamiliar with the CAS Disclosure Statement, a mapping between elements of cost described in that Disclosure Statement to those in the parametric model is recommended to ensure consistency between the two.

7.1.4 Federal Acquisition Regulations and Sundry Issues

7.1.4.1 Forward Pricing Rate Agreements

FAR 15.407-3 identifies regulatory requirements specific to forward pricing rate agreements (FPRAs). Where annual dollar volume and/or number of pricing actions warrant, regulation encourages that FPRAs be negotiated between a contractor and the Government to ensure rates are valid and available for pricing actions over a specified period of time. FPRAs are often established for direct and indirect rates, and are routinely used for CERs.

FPRAs generally include these elements:

- Clear definition of cost and non-cost elements pertaining to the CERs;
- Specified timeframe of applicability;

- Appropriate applications of the CERs;
- Processes for monitoring the CERs;
- Provisions for rescission by either party with proper notification;
- Specified cost reports to be furnished or made available at specified regularly cited intervals.

In situations where cost or pricing data are required, FPRAs are certified each time a specific pricing action is negotiated. When FPRAs are used for CERs, it is important to have monitoring procedures in place, based upon specific cost reports to made available at regular intervals.

The key to formulating the frequency of reports is to make certain that if the CERs are no longer valid, sufficient advance identification can be made to mitigate further windfall profits or losses by exercising the rescission provisions in a timely fashion. As such, it is essential to have effective processes for identifying any unusual events that may have a significant effect on the CERs, such as changes in production processes or company reorganizations. As an example, during the Parametric Estimating Reinvention Laboratory, one of the teams established a process for monitoring CER accuracy. The IPT defined a range of acceptability (or tolerance level) for each CER, and established processes to monitor CER accuracy on a monthly basis to identify any anomalies, such as CERs falling outside the defined range. The IPT analyzed these anomalies and identified follow-up activity to update or improve the CERs.

7.1.4.2 Format(s) for Submitting Cost or Pricing Data

FAR 15.408 (l - m), Table 15-2, *Instructions for Submitting Cost/Price Proposals When Cost or Pricing Data are Required*, contains guidance for preparing a contract pricing proposal with cost or pricing data. The FAR requires contractors to provide all information needed to explain the estimating process used, including detailed descriptions of (1) specific estimating techniques used, and (2) any judgmental factors. In addition, Table 15-2 instructions provide guidance related to the breakdown of proposed cost elements (e.g., materials, direct labor, and indirect costs). Again, as with more traditional techniques, parametric techniques can be used to generate costs in this format, or they can be used to generate cost breakdowns in varying formats such as a work breakdown structure (WBS). The solicitation clause FAR 52.215-20, *Requirements for Cost or Pricing Data or Information Other Than Cost or Pricing Data*, permits a contracting officer to tailor the proposal format as needed, although they are encouraged to use the contractor's format as much as possible.

When a contractor plans to use a complex parametric model to develop a proposal, and desires to use an alternative to the Table 15-2 format, they should work with their customer up-front to identify a proposal format that the customer is willing to accept. During the Parametric Estimating Reinvention Laboratory, one of the participants that used a complex commercial model included the following information as a supplement to Table 15-2 to support its basis of estimate (BOE).

- Background on the commercial parametric model used.
- Description of the calibration process (including identification of specific programs used).
- Results of the validation process.
- Application of the commercial model to the current estimate, including:
  - Identification of the key inputs used, including the contractor's rationale;
  - Basis for any adjustments made to the model;
  - Disclosure of comparative estimates used as a sanity check, to demonstrate the reliability of the parametric estimate.

7.1.5 Subcontracts

The treatment of subcontract costs and compliance with FAR regarding subcontract costs has been one of the most significant challenges to implementing parametric techniques in proposals. However, this is an issue for all estimating approaches. Therefore, it is imperative that the treatment of subcontract costs in the proposal and negotiation process is addressed early and an agreement is reached between the contractor and the Government.

FAR 15.404-3 defines cost or pricing data requirements specific to subcontracts. It states that a prime contractor is required to obtain cost or pricing data if a subcontractor's cost estimate exceeds $650,000[1], unless an exception applies. A prime contractor is also required to perform cost or price analysis on applicable subcontractor estimates to establish the reasonableness of the proposed prices.

Prime contractors are required to include the results of the analyses with proposal submissions. For subcontracts that exceed the lower of $10,000,000, or are more than 10 percent of a prime contractor's proposed price, the prime contractor is required to submit the prospective subcontractor's cost or pricing data to the contracting officer. If the subcontractor does not meet this threshold, but the price exceeds $650,000, the prime contractor is still required to obtain and analyze cost or pricing data but is not required to submit it to the Government.

Subcontractors should be responsible for developing their own estimates since they have the experience in pricing the specific good or service they will be providing. Subcontractors are in the best position to include the cost impacts of new events such as reorganizations, changes in production or software engineering processes, and changes in prices of key commodities.

[1] This is the current threshold at the time of this update. The contracting officer should check the current threshold.

For these reasons, it is a best practice for a prime contractor to obtain the necessary cost or pricing data directly from its subcontractors. Prime contractors can work with their subcontractors to streamline costs and cycle time associated with preparation and evaluation of cost or pricing data. This means that subcontractors can use parametric estimating techniques to develop their quotes, provided their models are adequately calibrated and validated.

The Government may decide that adequate evaluation of a prime contractor's proposal requires field pricing support (an assist audit) at the location of one or more prospective subcontractors at any tier. This may be based on the adequacy of the prime contractor's completed cost analysis of subcontractor proposals.

The prime contractor's auditor will also evaluate the subcontractor's cost or pricing submission. The prime contractor will advise the contracting officer if they determine there is a need for a Government assist audit. If the prime cannot perform an analysis of the subcontractor's cost or pricing submission in time for proposal delivery, the prime will provide a summary schedule with their proposal. That schedule will indicate when the analysis will be performed and delivered to the Government.

The following items generally indicate a need for a Government assist audit.

- The prime contractor's cost analysis is inadequate or is not expected to be completed prior to negotiations.
- The prime contractor's policies and procedures for awarding subcontracts are inadequate.
- There is a business relationship between the prospective prime contractor and subcontractor that is not conducive to independence and objectivity as in the case of a parent subsidiary or when prime and subcontracting roles of the companies are frequently reversed.
- The proposed subcontractor costs represent a substantial part of the total contract costs.
- The prospective prime contractor was denied access to the proposed subcontractor's records.

7.1.6 Earned Value Management System (EVMS) and Contractor Cost Data Reports (CCDR)

Due to the shrinking DoD budget and funding constraints cost growth on DoD cost type contracts has been one of the most significant issues within DoD. Through the use of parametric modeling techniques the Government can effectively evaluate the reasonableness of the contractor's estimate of total contract costs and perform "what if" analysis of what would happen if the schedule changes and or scope revisions occur.

Through the use of the data included in standard EVMS and CCDR reports, the Government can use this data to control costs on significant programs.

Parametric estimating tools can be used to evaluate many aspects of the contractor's assertions, as reported on its standard reports produces from their internal systems. For example, historical performance data included in the EVMS and CCDR report can be used to project expected future total costs as part of the evaluation of the contractor's estimate at completion.

Parametric estimating techniques can also be used to evaluate the feasibility of the contractor's assumptions relating to expected cost changes due to planned management action or program changes by performing "what if" analysis. In addition, due to the detailed nature of these reports and the level of data accumulation and analysis, these reports are very useful in validating and calibrating historical data and should be reviewed when developing a parametric model for future costs.

7.1.7 Best Practices

Properly calibrated and validated parametric techniques can comply with all Government procurement regulations. Establishing effective estimating system policies and procedures specific to the proposed parametric techniques ensures consistent compliance with the applicable statutes and regulations, provided they are successfully implemented and enforced through periodic internal reviews.

Using teamwork, IPTs, and addressing the best practices discussed in this chapter, contractors can comply with all Government procurement regulations while making parametric estimating techniques an accurate and reliable tool for streamlined estimating processes. For example, during implementation of a parametric-based estimating system, Government team members can provide feedback to the contractor concerning their expectations related to the estimating system disclosure requirements. In addition, the Government team members can work with the contractor to address any other regulatory concerns on a real-time basis so improvements can be initiated before actual proposals are submitted.

Due to the sensitivity of subcontract costs the treatment of these costs should be addressed early between the contractor and the Government, preferably as part of an IPT. In addition a MOU should be developed relating to the treatment and disclosure of subcontract costs. See Appendix F.

7.2 Parametrics and the Government Review Process

Parametric estimating techniques are evaluated as part of a contractor's estimating system to ensure accuracy and reliability of individual proposals. FAR instructs, as appropriate, "cognizant audit activities shall establish and manage regular programs for reviewing selected contractors' estimating systems or methods in order to reduce the scope of reviews to be performed on individual proposals, expedite the negotiation process, and increase the reliability of proposals."

In general, DCAA evaluates estimating systems to verify that they are based upon:

- Credible data;
- Sound estimating techniques;
- Good judgment;
- Consistent estimating approaches;
- Compliant practices and policies with law and regulation.

While the following Figure 7.1 captures the process flow for auditing both parametric estimating systems and proposals, often an IPT approach is employed by the Government, which includes not only DCAA, but also DCMA and the major buying activities thereby leveraging all available technical expertise such as engineers and pricing people with knowledge of product lines and software modeling.

In general, when evaluating parametric estimating systems, the focus is on:

- Credibility of databases, meaning currency, accuracy, completeness, consistency of availability over a period of time, and verifiability to source documentation;
- Methodologies used to perform data collection, analysis, calibration and validation in order to reliably predict costs;
- Policies and procedures established to enforce and ensure consistent application of appropriate parametric techniques.

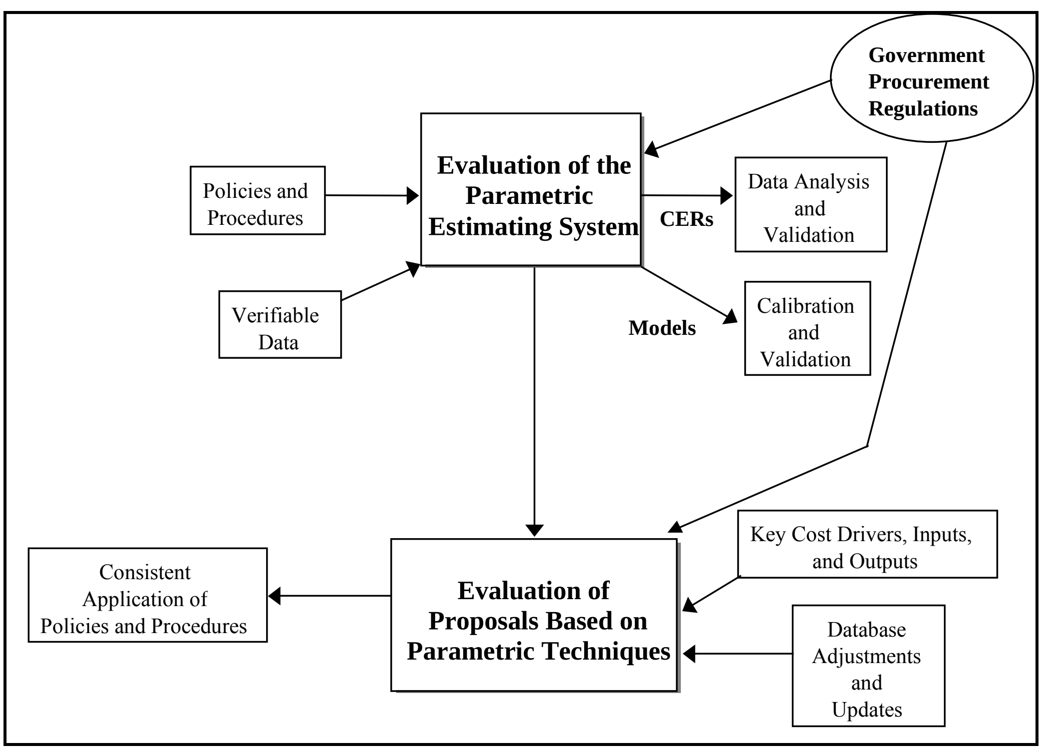


Figure 7.1 Audit Process Flowchart

When reviewing individual proposals, the focus should be on:

- Determining that the estimates were developed in accordance with established policies and procedures;
- Evaluating key cost drivers, inputs and outputs;
- Identifying and analyzing any adjustments to the database, CERs or input parameters.

7.2.1 General Estimating System Requirements

DoD policy mandates contractors have estimating systems that consistently produce well-supported proposals to provide an acceptable basis for negotiating fair and reasonable prices. Specifically, DFARS 215.407-5-70 provides guidance for conducting estimating reviews, and requires that not only all DOD contractors have acceptable estimating systems, but certain large businesses disclose their estimating systems as well. DFARS defines an "acceptable estimating system" as being (1) established, maintained, reliable, and consistently applied; and (2) producing verifiable, supportable, and documented cost estimates.

Further, DFARS requires an acceptable estimating system to:

- Assure that relevant personnel have sufficient training, experience, and guidance to perform estimating tasks pursuant to established procedures;
- Identify the sources of data, the estimating methodologies, and rationale used in developing cost estimates;
- Provide for consistent application of estimating techniques;
- Provide for the use of historical experience, including historical vendor pricing information, as appropriate;
- Require use of appropriate analytical methods;
- Require management review, including verification that the estimating policies, procedures, and practices comply with applicable regulation;
- Provide for internal review of, and accountability for, the acceptability of the estimating system, including comparison of projected to actual results, and analysis of differences;
- Provide procedures to update cost estimates in a timely manner throughout the negotiation process;
- Address responsibility for review and analysis of the reasonableness of subcontract prices.

Additionally, DFARS provides that significant failure to establish and maintain an adequate system as described above could result in the reporting of the noted deficiencies. Implementation and consistent use of appropriate, properly calibrated and validated parametric estimating techniques facilitates compliance.

7.2.2 Parametric Estimating System Requirements

7.2.2.1 Policies and Procedures

Auditors are charged with determining whether estimating system policies and procedures are established, available, and adequately address all significant parametric techniques employed by a contractor. Policies and procedures should include guidelines for:

- Determining when specific parametric techniques are appropriate;
- Collecting and normalizing data;
- Identifying logical relationships and analyzing the strengths of those relationships;
- Calibrating and validating parametric techniques, including identifying significant cost drivers, input/output parameters, and procedures for routinely validating the accuracy of the model;
- Developing estimates for proposals based on parametrics;
- Identifying and analyzing significant data between established cyclical updates to determine if out-of-period adjustments are required;
- Ensuring consistent application of parametric techniques;
- Ensuring personnel have sufficient training, experience, and guidance to perform parametric estimating tasks in accordance with established estimating processes;
- Performing internal reviews to assess compliance with estimating policies and procedures, including techniques for periodically assessing the accuracy of the parametric estimates.

7.2.2.2 Credible Data

Contractors are encouraged to use historical data, whenever feasible, as the basis of estimate. Technical representatives at both the buying command and DCMA may have specific knowledge as to the appropriateness of that data and be in a position to provide valuable technical support to DCAA accordingly. For example, actual costs may reflect gross inefficiencies due to initial engineering or manufacturing difficulties encountered, but since resolved.

Parametric techniques generally require the use of cost, technical, and other programmatic data. Figure 7.2 is an example of the types of data customarily collected.

Description	Examples
Cost data	Historical costs, labor hours
Technical Data	Engineering drawings, engineering specifications, weights
Programmatic data	Development and production schedules, number of units

Figure 7.2 Types of Data Collected

Whatever the nature of the data, it should be normalized so that it is consistent between data points by adjusting for differences due to inflation, learning, changes in scope of work, and other specific anomalies. However, the relevance of the data must be explained. The Government view of data normalization is discussed in Chapter 2.

7.2.2.3 Cost Estimating Relationships

The CER development process normally includes data collection, analysis (including normalization), and validation. In analyzing potential cost drivers, all reasonable alternatives should be considered, relying upon experience and published sources to identify them. From a contractor's perspective, not only should the relationships be logical and accurately predict costs based upon a sufficient number of data points for implementation and testing or validation, but whether the requisite data to be collected can be made readily available without undue expense.

From the Government's perspective, auditors and supporting technical reviewers need to evaluate and monitor significant CERs to ensure reliability and credibility as cost predictors by:

- Determining if the data relationships are logical;
- Verifying that the data used are adequate and verifiable;
- Performing analytical tests to determine if strong data relationships indeed exist, using judgment to establish an acceptable range for accuracy;
- Ascertaining whether CERs are used consistently with established policies and procedures, and comply with all Government regulations.

7.2.2.4 Complex Models

Auditors and technical reviewers will concentrate on calibration and validation techniques employed, as well as the corresponding policies and procedures, keeping in mind key inputs and outputs of the model being adapted.

The Government View of Calibration

As previously stated, calibration is the process of setting up a complex parametric model so that it incorporates the contractor's cost and product history. The parameters of a complex commercial model, when purchased, are set to identified default values. Accordingly, the calibration process permits a company to adjust those values so that the model's output reflects the contractor's specific business environment and practices.

Proper calibration of a complex model has a significant impact on ability to accurately predict costs. For this reason, complex models should be calibrated before they are used as a basis of estimate. Consequently, a contractor will need to establish policies and procedures that discuss its calibration methodologies, including information on the model's significant cost drivers and associated input parameter values. Figure 7.3 provides an example of key inputs for complex parametric hardware and software models.

Commercial Parametric Models Examples of Parameters (Inputs)	
Hardware Model	**Software Model**
Weight	Software size
Quantity	Development language
Engineering complexity	Software process maturity level
Manufacturing complexity	Software tools
Schedule	Personnel capabilities

Figure 7.3 Sample Inputs for Complex Parametric Models

For the contractor, data collection and analysis is generally the most time-consuming part of the calibration process thereby making it cogent that the calibration methodology be defined prior to collection of data. A significant portion of that data should be obtained preferably from the organization's information systems, while other data, such as technical data, are usually obtained from a variety of sources ranging from manufacturing databases to engineering drawings. Otherwise, contractors may interview technical personnel to obtain information not readily available, such as information germane to a specific product or process. The data are then normalized.

Complex models generally have their own classification system for cost accounts, and as a result, companies must establish a mapping procedure to properly relate their cost accounts to those used by the models for accuracy of predictions and to preclude potential noncompliance with the Estimating System and/or Cost Accounting Standards Disclosure Statements. Additionally, contractors should document any adjustments made to the data, including assumptions and associated rationale, during mapping.

Once data are collected, normalized, and mapped, they are entered into the model. The model then computes a calibration or correction factor that is applied to that data. The result is a complex model adjusted to represent the organization's experience, or "footprint". As such, contractors must document the calibration process fully, including the key inputs, input parameters, calibration assumptions, results of interview questionnaires, i.e., names of people interviewed, dates, information obtained, as well as cost history, and the calibration estimate. Any changes in these calibration factors over time must also be documented.

Auditors and technical reviewers when evaluating the calibration process will need to note whether:

- Credible data was used;
- The data points used in calibration are the most relevant to the product being estimated;
- Adjustments made to the data points were required and documented to evidence reasonableness and logic;
- Data outside the normal update schedule are appropriately analyzed to determine if out-of-period updates are needed;
- Historical data used for calibration can be traced back to their sources;
- An audit trail for the entire calibration process is documented;
- Processes used to normalize and map data ensure the underlying assumptions are logical and reasonable;
- Key input values and complexity factors generated by the model are reasonable;
- Parameter values are comparable to values derived using a variety of sources such as ranges specified in policies and procedures, values used in prior estimates, ranges recommended in commercial model manuals, and results of prior calibrations;
- Procedures for calibrating to the most relevant data points, including underlying assumptions and analysis of alternative points, exist and are adequately addressed;
- Use of other than historical data points is documented;
- Policies and procedures and actual practice successfully identify, analyze and process significant out-of-period changes resulting in updates to maintain currency and completeness.

Validation

As previously stated, validation is the process or act of demonstrating a model's ability to function as a credible forward estimating tool, and therefore predict costs with accuracy. It ensures that calibration process was properly performed and documented, the estimating procedures were established and consistently enforced, and key personnel were adequately trained.

Contractor and Government reviewers can perform benchmark tests when determining the ability of a model to accurately and reliably predict costs. For example, where sufficient historical data exist, an independent data point can be withheld from the calibration process and used as a test point in the calibrated model, or the parametric estimate may be compared with one developed using other estimating techniques. Validation is an on-going process, and procedures should be in place to routinely monitor parametric estimating techniques to ensure they remain credible.

Accuracy is a judgment call, since no standard or pre-defined level of exactness exists for any given model. Generally, the more significant the cost element being estimated, the higher the degree of accuracy expected. When establishing an acceptable level of accuracy, reviewers should consider the uncertainty associated with alternative estimating methods that could be employed. If the level is lower than desired, additional data analysis is probably necessary. When evaluating and monitoring accuracy levels of complex models, the focus is most profitably directed on the model's key cost drivers. As a rule of thumb, there are a few parameters that drive the cost for any given product line (e.g., weight, complexities, software size), and should thus be the focus of that in-depth evaluation.

7.2.2.5 Company Developed Models

Company developed models, like commercial ones, consist of CERs, other mathematical relationships, and their associated logic, as well as programmatic inputs such as system description data to generate outputs such as costs. They are generally built for a specific purpose that commercial models cannot satisfy, and range in complexity from simple spreadsheet applications to more advanced paradigms. Company developed models are self-calibrated in that they are based on an organization's own historical information.

Evaluation of company developed models should focus on the points previously discussed in this chapter, as well as a few additional areas including:

- Costs and benefits of developing and maintaining a company developed model;
- Information technology (IT) controls established to monitor system development and maintenance, and to ensure continued integrity of the system;
- Model testing and verification to ensure it produces the expected results.

Auditors and supporting technical personnel, including software specialists, should evaluate any cost-benefit analysis performed prior to the development effort, as well as the model's IT controls. The purpose of reviewing the design and development of new IT systems is to confirm it incorporates the economic, efficient, and accurate execution of management policies in an auditable and controllable environment.

Some key audit considerations to corroborate are:

- Documentation exists that thoroughly addresses the parametric model, including types of processing performed by the model, the data processed, the reports generated, and user instructions;
- Sufficient testing is performed to demonstrate the model functions as required, and produces credible results;
- Proper controls are in-place to monitor changes to the parametric model, such as the integrity of model and security controls such as access to the model;
- Personnel are trained and experienced in performing model development.

Audit and other review efforts expended in evaluating IT controls should be commensurate to the combined costs of developing and maintaining the model, the significance of the costs generated, and the complexity and amount of independent analysis performed internally by the contractor to evaluate its soundness. Companies generally involve their internal audit staff, as well as Government evaluators, during the various phases of model development and testing to assess and demonstrate its capabilities.

7.2.3 Parametric-Based Proposals

The review of a proposal that has cost based on parametric techniques should be relatively simple and straightforward, provided (i) it is based upon established parametric estimating policies and procedures (ii) such policies and procedures have been deemed adequate and compliant with procurement law and regulation, and (iii) it is properly calibrated and validated. As such, emphasis is placed upon determining that estimates are consistent with those policies and procedures, and any deviations are adequately justified in writing.

To facilitate Government review, proposals should contain sufficient documentation that Government reviewers can use to evaluate the reasonableness of estimates. Dependent upon the model or parametric technique being used, a basis-of-estimate (BOE) for a parametric cost estimate should include the following types of information:

- A description of the program, products, services, or individual cost elements being estimated.
- A description of the commercial or in-house developed model or CER(s) used in developing the estimate.
- All cost drivers considered in preparing the estimate (cost and non-cost parameters).

- The types of materials (raw, composite, etc.) and purchased parts with procurement lead times required to complete the tasks being estimated.
- The types of direct labor and/or skill mix required to perform the tasks being estimated (e.g., manufacturing, manufacturing engineer, software engineer, subcontract manager).
- The time-phasing of the use of resources in performing the tasks being estimated (i.e., matching the cost of materials, labor, other direct costs, and indirect expenses with the periods – weeks, months, or years – the resources will be used).
- The estimating method, rationale, assumptions, and computations used to develop the estimate. Estimates should also include a comparison of the current estimates to the historical program, including explanations of any adjustments made to historical cost or operational (non-cost parameters) data (e.g., use of complexity factors).
- Other elements as required by management or customer instructions.

In the case of CERs, the BOE should explain the logical relationship of all cost-to-cost and/or cost-to-non-cost estimating relationships used in the estimate. It should include (i) a description of the source of historical data used in determining the dependent and independent variable relationships and its relevance to the item or effort being estimated, (ii) a description of the statistical analysis performed, including the mathematical formulas and independent variables, and an explanation of the statistical validity, and (iii) any adjustments made to historical data to reflect significant improvements not captured in history, such as changes in technology and processes.

When a commercial model is used in preparing an estimate, the BOE should describe the estimating model used, and identify key input parameter values and their associated rationale, as well as model outputs. The BOE should describe how the model was calibrated, that is, describe the process for developing factors that adjust the models computations to more closely reflect the contractors' specific environment. Auditors and other Government reviewers will assess the database's validity to ensure currency, accuracy, and completeness by checking that the most current and relevant data points(s) were used for calibration. Accordingly, identification of the historical database in the BOE is essential. Another key assessment is determining how the processes and technologies will be used on the programs being estimated as compared to the same for those programs contained in the calibration database. Use of technical support at DCMA and the buying command by DCAA may be appropriate. In addition, the BOE should describe how the model has been validated.

Additionally, contractors may submit proposals for forward pricing rate agreements (FPRAs) or formula pricing agreements (FPAs) for parametric cost estimating relationships to reduce proposal documentation efforts and enhance Government understanding and acceptance of the estimating system. The basis of estimate should include the information described above for CERs and should clearly describe circumstances when the rates should be used and the data used to estimate the rates must be clearly related to the circumstances and traceable to accounting and/or operations records.

Also, with the advent of reorganizations and process improvement initiatives such as single-process initiatives, software capability maturity model improvements, and technology improvements, adequate procedures should be in-place for quantifying the associated savings and assuring their incorporation into the estimates. Often such changes are reflected in decreasing complexity values or downward adjustments to the estimate itself. Regardless of cause, Government reviewers will evaluate any significant adjustment, including the pivotal assumptions and rationale, to determine if it is logical, defensible, and reasonable.

7.2.4 Best Practices

Here is a list of best practices related to the Government review process.

- While audits of proposals using parametric estimating techniques should be similar to those performed using other estimating methods (e.g., bottoms-up or analogous estimates), emphasis should focus on evaluating the policies and procedures of the estimating system. Contractors rely on these procedures to produce well-supported proposals that are acceptable as a basis for negotiating fair and reasonable prices.
- Sufficient historical data relevant to the current environment often does not exist, necessitating the use of other data points for calibration. Accordingly, auditors should use judgment when evaluating data used for calibration and validation, while contractors need to establish formal data collection practices to ensure effective use of parametric techniques thereby better controlling the time consuming data gathering process.
- When validating the capability of a particular model's ability to predict costs, it is preferable to use an independent data point such as actual history not used in the model's calibration. However, due to limitations, independent data points may not be available. Therefore, judgment is required when considering and evaluating other validation approaches such as comparisons based on independent estimates performed by other estimating groups within the contractor's organization, and prepared using other conventional estimating techniques.
- Statistical measures are not the only criteria to be used in determining the validity of CERs so that a variety of tests should be performed, with no one test disqualifying it. Other factors to be considered include the logic of the relationships, soundness of the data, and adequacy of the policies and procedures, as well as the assessed risk associated with the CER versus the effectiveness of estimating techniques previously used to predict those costs.

- Since there is no pre-defined accuracy level, by default it is a matter of judgment, attendant with a degree of uncertainty and risk as with any estimate. Therefore, it is integral in maintaining integrity to consider the magnitude of the costs being estimated, as well as the ability of alternate techniques to predict them. Concurrently, it is important to have a monitoring process in place to identify areas of concern that warrant additional scrutiny and contribute towards continuous improvement.
- To effectively evaluate a contractor's parametric estimating system, auditors and other Government reviewers should have a good understanding of a company's estimating practices, the specific techniques and/or models used, and the characteristics of the database. As such, training may need to be provided by the company, or commercial-model vendors, or other available sources. Ideally, IPT training provides all team members with a common understanding that can be used to develop acceptable calibration and validation approaches.
- Many contractors implement parametric cost estimating techniques such as parametric BOEs and establish IPTs, including representatives from the company, the buying activity, DCMA, and DCAA. This should be done at the beginning of the developmental process for either the system or a specific proposal, so all members can provide feedback on the approaches that will be used to estimate costs, e.g., collecting, normalizing, and analyzing data, as well as calibrating, validating, and monitoring the parametric estimating system and techniques. By doing so, Government IPT members can provide real-time feedback that reflect the Government's expectations germane to the estimating system policies and procedures, calibration and validation criteria, and other significant evaluation criteria. At the same time, formation of an IPT increases the confidence of all parties in the parametric estimating process, the model, and resulting estimates. Additionally, establishment of a sound database, the single most time consuming process in developing a parametric tool, is a prerequisite for effective implementation.

7.3 Technical Evaluation of Parametrics

This section delineates those technical tools and techniques DCMA commonly uses to develop and evaluate parametric proposals, including statistical evaluation, data collection and analysis, calibration, and validation. Flowcharts provide a framework for discussing the successful evaluation of parametric techniques, including CERs and complex models, whether commercial or contractor specific. Additionally, the discussion is more applicable to cost analysis rather than price analysis in that the latter requires far less data for analysis by its very nature, and thus would not prove cost effective.

7.3.1 Authority to Evaluate Parametric Techniques

Uncertainty exists as to whether the Government accepts the use of CERs and parametric models to develop proposals submitted to them. Specifically, FAR 15.404-1(c)(2) states, "The Government may use various cost analysis techniques and procedures to ensure a fair and reasonable price, given the circumstances of the acquisition. Such techniques and procedures include the following:

(i) Verification of cost or pricing data and evaluation of cost elements, including

(C) Reasonableness of estimates generated by appropriately calibrated and validated parametric models or cost-estimating relationships."

Accordingly, properly calibrated and validated parametric techniques are acceptable as a basis for estimate, and therefore are subject to technical evaluation by the Government.

7.3.2 Cost Modeling Process for CERs

This section provides some technical guidance for the review of CER estimates. The flowchart in Figure 7.4 describes the general framework of CER development.

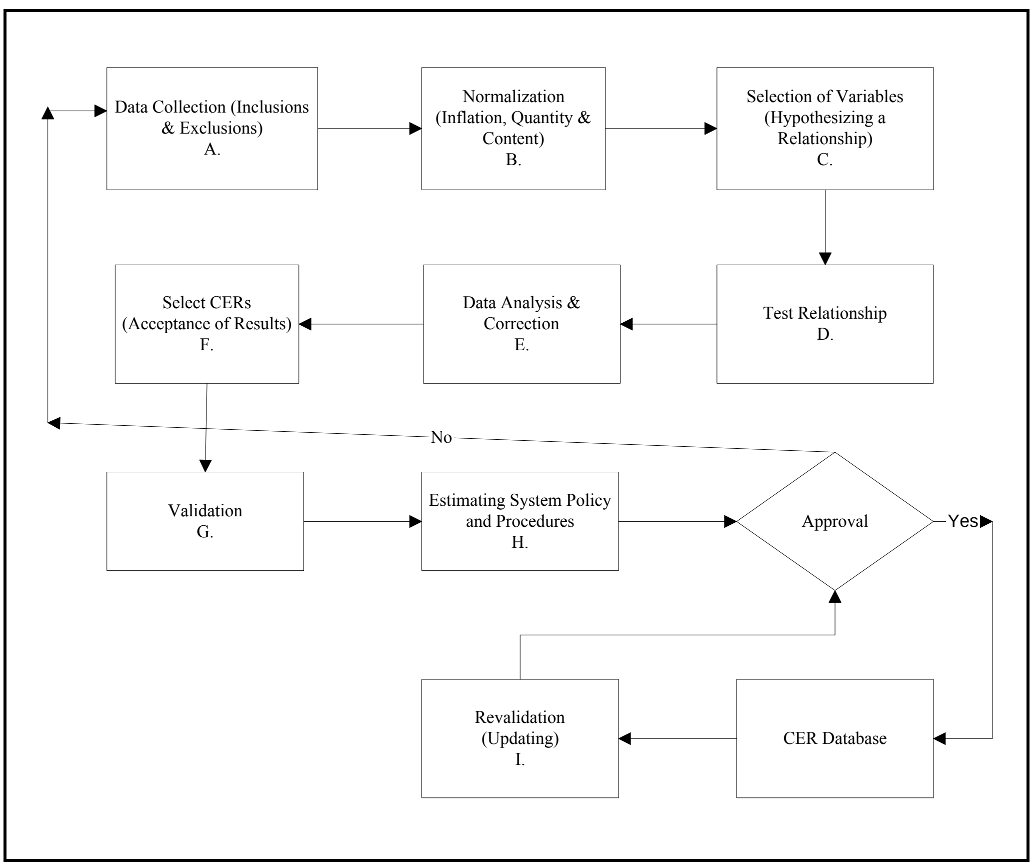


Figure 7.4 Typical CER Development Process

7.3.2.1 Step 1: Data Collection (Inclusions and Exclusions)

When establishing or reviewing an organization's data collection procedures, the decision to either omit or include a given point of data should be determined based upon its predictive value. For example, when noting a five-year trend of consistent decreases, should the latest single point reflecting an increase be designated an anomaly and hence omitted, or have circumstances changed so markedly that only that last point of data presents the probable future for that cost driver? Such issues must be identified and adequately analyzed.

Moreover, it is may be inappropriate to dismiss outliers out-of-hand solely on the basis that they lie far from the statistical mean. Investigation is always required. Generally, if the proximate cause of the suspect data points is likely to reoccur, the datum should be accepted. For instance, assuming that an outlier is the result of a work stoppage, a determination as to whether to omit or include it depends upon the likelihood of a work stoppage occurring during the proposed period of performance.

Another consideration that arises when collecting data concerns how much historical data should be collected. The natural tendency is to include as much as possible. Although additional data points can provide statistical confidence, this may be illusory. For example, the older a data point is, the more likely it is not to represent current conditions due to changes that occurred in the configuration and/or in the business environment. However, this is not to say data should be rejected solely because of age.

7.3.2.2 Step 2: Normalization (Inflation, Quantity and Content)

See Chapter 2 for a discussion on this topic.

7.3.2.3 Step 3: Selection and Testing of Variables (Hypothesizing a Relationship)

For the sake of elucidation, let us consider the propriety of a simple CER for estimating inspection costs where: inspection, *i,* is the dependent variable; manufacturing labor and material, "*l*" and *"m,* are the independent variables; and, *a* is a constant. It then could be postulated that inspection costs are driven by labor, with $i = (a)(l)$, or to material, with $i = (a)(m)$, or to the sum of manufacturing labor and material, with $i = a(l + m)$. Inspection costs are anticipated to be less than manufacturing and/or material so that $a < 1$.

Each of the above alternative hypotheses will need to be tested using the available data, and if need be the problem redefined so as to deal with the process and final inspection costs separately. The results of these alternative approaches will need to be tested or analyzed to determine whether they produce significant differences in the cost estimate, as well as their impact over different time periods. A good predictor should minimize the variability and, hence, lower the risk associated with the estimate.

The inspection CER discussed above is an example of a linear relationship. However, an exponential ($i = a * l^2$) or logarithmic ($i = a + 2\log l$) relationship could have been postulated. Often linear relationships are favored, because they are easier to evaluate and comprehend. Nonetheless, it is important not to arbitrarily rule out non-linear relationships. Improvement or learning curves are an example of a generally accepted exponential relationship. Additionally, for the above inspection scenario, an analyst should have postulated and tested to determine whether the CER will be used to estimate both initial and final inspection costs jointly or separately for the greatest accuracy.

7.3.2.4 Step 4: Data Analysis and Correlation

In determining whether a linear or non-linear relationship is best for a CER, the correlation of the independent and dependent variables for each possible formulation will need to be evaluated to determine the one with the best correlation. It is paramount that the relationship chosen is logical. A CER with the best correlation is not useful if its underlying relationship does not make sense.

More is not always better. In some cases, testing through regression analyses show that when selecting the variables for the development of a CER, based upon the correlation between the attributes of the item being estimated and the corresponding historical data points, the addition of more variables does not always increase estimating accuracy.

7.3.2.5 Step 5: Select CERs (Acceptance of Results)

Now let us assume inspection costs for the most recent lot equaled 18 percent of manufacturing labor, $i = (0.18)(l)$. Both estimators and evaluators need to ask themselves whether that formula best represents the CER for all inspection processes and the results are deemed sufficiently valid and accurate for estimating purposes to be declared acceptable. To make these determinations, many considerations need to be addressed. For example, what was the experience for the prior lot? Was this value significantly higher or lower than the same for other historical lots? Is there a trend, and if so, does the estimate consider it, or have significant changes occurred so as to preclude the propriety of using that historical data?

In deciding whether a CER should be program specific, estimators and technical evaluators need to determine not only what makes the best technical sense, but also if a facility average has been used historically In such instances, the establishment of a program specific CER without effecting a revision to the CAS Disclosure Statement may result in a noncompliance with Cost Accounting Standard 401 for CAS-covered contracts.

Also, before accepting a CER for usage, it should ascertained whether technological changes indicate any need for further adjustment in the base data and if all of the data has been normalized appropriately. As importantly, the attendant risk associated with any lack of accuracy should be determined. Obviously, the smaller a job or cost element being addressed, the less is the corresponding risk. Accordingly, the larger the dollars determined at risk, the greater is the need to thoroughly evaluate the proposed CER prior to acceptance and usage

Finally, it is prudent to develop an independent estimate to determine whether to accept a CER. The general rule is to accept the CER if the two estimates are reasonably close; if not, another approach can be tried, such as preparing a detailed analysis of the divergent estimates to determine whether either of them contains flaws in logic or specious data.

7.3.2.6 Step 6: Validation

Validation is the process of demonstrating that the CER or cost model accurately predicts past history, or current experience. This entails demonstrating that the data are credible, and that the relationship(s) is logical and correlates strongly. In determining if a CER or model is a good predictor for future costs, its accuracy needs to be assessed. As was previously discussed, the best technique is to use independent test data, that is data not included in the development of the CER or model. In limited data situations, however, flexibility is needed to develop alternate approaches.

Regardless of approach, good judgment is a requisite to determine an acceptable level of accuracy because there is no recognized standard level for CERs or more complex models. In general, CERs and cost models should be at least as accurate as the prior estimating technique relied upon.

7.3.2.7 Step 7: Estimating System Policies and Procedures

Contractors that use parametric techniques for proposals submitted to the Government or higher-tier contractors need to establish policies and procedures that thoroughly define the estimating methodology employed pursuant to the estimating system criteria described in Section 7.2 to ensure regulatory compliance.

7.3.2.8 Step 8: Revalidation (Updating)

The frequency of updating the database is another key consideration. Risk exists that the projected results may differ significantly between a projection not using the most recent data, and one incorporating said data. At the same time, there is a trade-off between the cost of continuously updating the database versus the expected benefits of using more current and complete data.

For this reason, FAR 15.406.2(c) permits for the use of cut-off dates. This regulation allows the contractor and the Government to establish defined cyclical dates for freezing updates. Some data are routinely generated in monthly cost reports, while others are produced less frequently. Accordingly, in establishing cut-off dates, the difficulty and costliness associated with securing the requisite data must be considered in tandem with the costs of actually updating the database.

Annual updates, for instance, may be appropriate for elements that involve a costly collection process, provided the impact in not updating more frequently remains insignificant. It follows that an annual update of some or all of the data implies that those portions of it may be as much as eleven months old at the time of consummating negotiations, which may be deemed an acceptable risk under an approved estimating system.

As such, a contractor's procedures will need to specify when updates normally occur, and indicate the circumstances and process for exceptions. Also, contractors need to have procedures established to identify conditions that warrant out-of-period updates. For example, a contractor may need to update its databases outside the normal schedule to incorporate significant changes related to such issues as process improvements, technology changes, reorganizations, and accounting changes.

7.3.3 Projecting Beyond the Historical Range

Statisticians state that an analyst should never make a parametric estimate using inputs which are outside the range of data used to build a CER or complex model. In cost estimating, however, this is usually an impossible rule to observe, since the estimator's goal is to extrapolate from the known to the unknown. However, when projecting beyond the data points the estimator needs to ensure the cost drivers have been validated. Learning curves, for example, are usually projected beyond the limits of the data, as are estimates for new product lines and projected costs after reorganization. Three different cases involving extrapolation for learning curves are discussed in the following sections.

7.3.3.1 Existing Products

It is necessary to first understand the theory behind a given parametric formulation before determining the limits involved in projecting outside its data range, and the associated risks. For the commonly used learning curve, the reduction in hours projected as more units are produced results from a combination of operator learning, more efficient use of facilities, and production line improvements possible with increased production rates. Other factors, such as improved training and supervision, can favorably affect learning, but regardless of causation, there is always a limit to the amount of improvement that can be achieved.

In some cases, there can actually be a loss of learning such as when production is disrupted, or a production line reaches full capacity (based on current manufacturing processes). In the case of the latter, older or slower equipment may be used and/or a second shift established to expand capacity.

Either remedy involves absorption of additional costs per unit of production, which is expressed as a loss of learning. For example, if additional operators must be hired, then the new operators begin at unit one on the learning curve. To some degree, learning may take place at a faster rate for the newer group than for the initial group, due to lessons learned, process improvements, and the resulting training and mentoring. However, this cannot be assumed.

Nonetheless, learning curves may be appropriately used, provided the projected efficiencies proportionately impact both the dependant and independent variables, as corroborated through validation testing.

7.3.3.2 New Product Lines

Projecting for continuous production under existing product lines is somewhat standard, assuming no complications exist. Parametric tools allow the abililty to perform "what-if" analyses. The first step is to plot the data upon normalizing it so that it approximates a straight line (such as time series, semi-log, or log-log). The line is then continued to the midpoint of the proposed effort. The resulting value is the appropriate estimate. However, when estimating new business or product lines, it is necessary to "bend the line" or travel up the line an undetermined distance to reflect a loss of learning.

Determining how to do this is difficult, even for the experienced estimator and evaluator because these determinations are subjective and open to conflicting expert opinion. In contrast, an inherent advantage of using parametric models is the built-in capability through the database to project the impact of developing new products. Regardless, the first step is to define the problem by isolating the areas that are new and require special attention from those that are a continuation of existing effort. Many CERs remain valid under this approach.

7.3.3.3 New Business Methods/Organization

In times of downsizing or mergers, reorganizations are common, and may result in accounting changes, including new structural descriptions for the direct and overhead costs. For CAS-covered contractors, such changes should be reflected in revisions to the CAS Disclosure Statements, a good resource for Government evaluators when taking a systemic approach to a contractor's parametric estimating system. Accordingly, under such circumstances the cost history would no longer be directly relevant without making appropriate adjustments to the data.

In many instances, new CERs may be required to estimate costs that were formerly overhead. Estimators and evaluators should not accept assurances from the contractor's top management that there is no cost impact, and continue to project as before.

The details of a CAS Impact Statement corresponding to the proposed revisions of the Disclosure Statement may indicate otherwise. Accordingly, it is strongly suggested that the historical data be normalized to convert it into the new format. Successful validation of the conversion can then be corroborated by testing the CERs or the model using a small project that has already been completed under the revised accounting system.

However, since it is not always possible to wait until a project is complete, sometimes the change will need to be tested in advance on sample proposals. Only in the case of a major impact would this be done. However, should it be achieved through an IPT, its members will have confidence in the normalization process used.

7.3.4 Breaks in Production

As was previously discussed, a break in production results in the loss of learning for both operators and supervision. Further, experienced operators may no longer be available when production resumes thereby necessiating the use of other personnel. Even with gaps in production with experienced personnel, some loss of learnng is expected. Additionally, if the break is sufficiently long, the line may be dismantled and require reconstruction necessitating the development of new method sheets and routings, as well as the replacement of tooling. In fact, after a long break, resumption may resemble the start-up of a new program. Perhaps this area involves more judgment than most for the experienced estimator and evaluator due to the low frequency that it is encountered during the typical work lifetime.

7.3.5 Personnel Reassignments and Relearning

Relearning is one of the consequences of a break in production, but it also occurs routinely without a hiatus, in which case it is relatively easy to handle. To some extent, the condition of personnel changes is an ongoing one and is already accounted for in the data. At any point in time, the production crew is likely to be a mix of new employees and seasoned hands. Yet, it is assumed all personnel at any given point have been associated with the program since its inception.

This paradigm does not create a problem unless the combined number of new and returning personel is higher than usual. A new person starting with no experience on the product line is considered to be at unit one of the learning curve. Accordingly, if one hundred units have already been built, that individual always will be short by that amount of accumulated learning. However, the impact of having fewer units of accumulated experience diminishes with time, but in theory, the new hire never performs as well as the original crew. In practice, this relative lack of experience becomes so inconsequential that parity is in effect achieved after a certain period: learning takes place at an accelerated pace for new operators during the initial phases, while concurrently improvement for experienced personnel continues at such a diminished, inconsequential pace that it is said to plateau or level off.

In parallel, a person returning to the program already has familiarity with the job so that acclimation generally occurs even more quickly than for a new hire. For such an individual, a possible method is to calculate learning at a greater than the normal pace until there is no notable residual loss of learning.

Figure 7.5 illustrates different learning concepts: continuous learning, a new operator brought in at unit 100, and an operator returning at unit 300 (the data are all notional and not based on actual programs). For routine production on established product lines not in continuous production over several years, a 90 percent curve is not unusual for labor hours. Relearning for labor on new product lines may be calculated on an 85 percent slope until it intersects the continuing operator's performance. However, specific historical company or industry-wide data, should take precedence over these rules of thumb, unless changing circumstances preclude applicability.

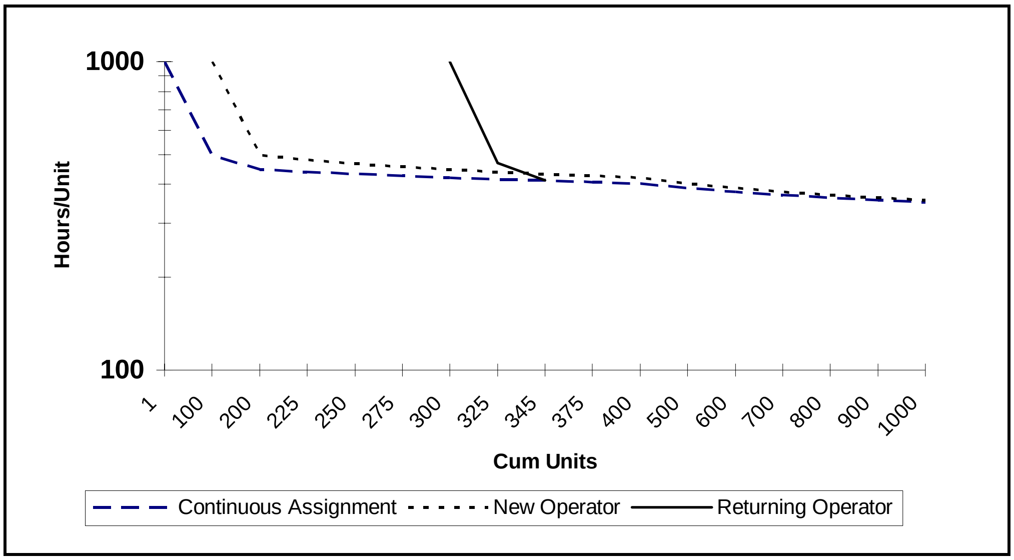


Figure 7.5 Different Learning Formulations

7.3.6 Best Practices

7.3.6.1 Accuracy Assessments

Limited accuracy may be acceptable for minor cost elements. However, major cost elements should be as accurate as possible, and will be subject to greater scrutiny. Whenever the results in any part of the evaluation process are questionable, alternative methods and hypotheses need to be considered. If the alternatives produce results similar to those under review, the accuracy of the modeling process is confirmed. If the results are different, then further examination is warranted.

7.3.6.2 Limited Data and Seemingly Poor Correlation

The use of traditional statistics, and established "accepted statistical criteria," is based on large sample sizes, often reflecting hundreds of observations. For many parametric applications, however, there are seldom more than five data points available to comprise a homogeneous set. Combining data points from diverse programs through data normalization may improve the size of the set, but typically the number of observations is limited to 25 to 30 points.

In a small data set, one aberrant point can dramatically skew statistical results thereby indicating poor correlation, as measured by the coefficient of correlation (R) or coefficient of determination (R^2). Neither of these coefficients constitutes the single best measure of a CER's validity, since they only represent the degree of linear fit (between independent and dependent variables), thus precluding their usefulness for applications with non-linear relationships. Nonetheless. both indicate a percentage of variation explained by the mathematical relationship used, which says nothing about the dispersion of the data or the likelihood that the estimate is correct. Rather variance or standard deviation measures dispersion, while confidence intervals measure likelihood. Consequently, to accept or reject a mathematical relationship based solely on its measure of correlation is incorrect. For example, a CER whose correlation is only 50 percent may be acceptable for a new program where the alternative is engineering judgment, and limited experience on which to base it.

The following questions represent a more appropriate hierarchy of concerns for evaluation.

- Is the relationship logical?
- Is there a good data collection system?
- Have statistical tests been applied properly?

Any software package that includes regression analysis applications can satisfy the last consideration. However, the first two questions are non-statistical in nature, and more important. Accordingly, if the collected data cannot be demonstrated to measure what it is purported to measure through the proposed relationship no amount of statistical analyses will foster confidence and trust in its usage.

7.3.6.3 Establishing an Implementation Team

IPTs, comprised of all the major parties such as DCAA, DCMA and the buying activity, facilitate the evaluation, negotiation, and implementation of a parametric system and use of BOE proposals. The IPT process allows for all stakeholders to:

- Understand the complex CERs and sophisticated models;
- Resolve problems, objections and obstacles upfront;

- Secure buy-in of the parties through the cumulative accretion of confidence in the end results as the process unfolds.

A disciplined management structure is needed to facilitate the implementation process. In addition, high-level endorsement from both the Government and the contractor is essential. Support must continue through all phases of the project, and early teaming with the Government is critical to its success. Also, the implementation of parametric estimating techniques should be facilitated and made uniform through the establishment of adequate policy and procedures. See Appendix J, Establishing a Parametric Implementation Team, for additional discussion on this topic.

7.3.6.4 Use of Specialists

During the implementation of parametric estimating systems, the use of experts in statistics and/or the given model being addressed may be an expedient and definitive in resolving challenges and conflicts. It also serves to make estimators, evaluators, and auditors more familiar and comfortable with parametric tools. For Government evaluations and negotiators, the use of specialists should be in consonance with local procedures and policies, if they exist. Otherwise, the need should be based upon judgment, and the identification of specialists or prospective specialists by referral or some other rational means.

7.3.6.5 Specialized Training Required

While evaluating CERs and models does not require a statistics expert, having a solid understanding of the basic mathematical concepts is critical including underlying probability and statistical measures. A summary knowledge of models and modeling is required. Further, if a particular model or technique will be used for proposals, it is advisable that joint training be provided to the members of the implementation team to establish a common understanding.

CHAPTER 8

Other Parametric Applications

Parametric techniques can be used for a variety of applications. This chapter provides a brief overview of things to consider when developing and implementing a parametric estimating model, implementation examples, lessons learned, descriptions of general and specialized applications where parametric estimating techniques can be used, and examples of where and how these techniques have been implemented.

More information about the examples shown in this chapter can be found on the web sites of the model developers or the International Society of Parametric Analysts and Society of Cost Estimating and Analysis. See Appendices A and D.

8.1 Tailor Applications to Customer Needs

The most important element for a successful parametric application is to involve all affected parties early in the development, testing, and implementation of the parametric model. The needs of both internal and external customers must be considered and coordinated during the entire process to ensure that the needs and concerns of all parties have been addressed.

Some effective tools that can be used to complete this process are as follows.

- Use integrated product teams (IPT). The Parametric Estimating Reinvention Laboratory demonstrated that the use of IPTs is a best practice for implementing, evaluating, and negotiating new parametric techniques in an estimating system. An IPT usually includes representatives from the contractor's organization as well as representatives from the contractor's major buying activities, DCMA, and DCAA.

  Using an IPT process, team members provide their feedback on a real-time basis on issues such as the calibration and validation processes, estimating system disclosure requirements, and Government evaluation criteria. By using an IPT, contractors can address the concerns of Government representatives before incurring significant costs associated with implementing an acceptable parametric estimating system or developing proposals based on appropriate techniques. The Parametric Estimating Reinvention Laboratory also showed that when key customers participated with the IPT from the beginning, the collaboration greatly facilitated their ability to negotiate fair and reasonable prices for proposals based on parametric techniques.

- Consider customer's requirements. Coordinate the specific requirements for known and expected customers to ensure that the model has the flexibility to provide data and estimates in the format required by the customer. A model that is not flexible or unable to meet the customer's requirements will not be used by the customer and renders the model effectively useless.
- Provide training to all customers. Training ensures the customer is aware of how the parametric tool was developed, what type of costs it is estimating, and help to ensure their "buy-in" on use of the model.
- Obtain a memorandum of understanding (MOU). Based on the results of the Parametric Estimating Reinvention Laboratory, it is recommended that a MOU be developed between the contractor and the customer to ensure that there is an agreement on what type of data will be provided to meet the customer's needs.

8.2 Considerations When Applying Tools

When applying the tools used in parametric estimating models, both the contractor and customer should consider the following items.

- Has there been a significant change in the underlining assumptions and data that was used to calibrate the model which may require the model to be adjusted or a new model developed?
- Has the parametric model been tested recently to ensure it is still providing accurate estimates?
- The materiality and risk associated with the estimate if the model provides an inaccurate result.
- Does the contractor organization have an adequate estimating system?
- Has the parametric technique been appropriately calibrated and validated?
- Were the assumptions used in the estimating process reasonable?
- Are the inputs to the parametric model appropriate?
- Were any significant adjustments made to the parametric technique/model?
- Are there any indications that the database requires an out-of-period update?
- Are the outputs from the parametric model realistic?

8.3 When and How to Use the Various Tools and Techniques

It's important to understand that the development and use of parametric tools requires an investment in time, money, and other resources. At times that investment can be sizable. If an organization is planning to develop a set of simple CERs to apply to estimates in their proposal system, the benefit to cost ratio will probably be excellent.

Many complex parametric models, on the other hand, require an organization to invest in model development or licensing fees. Complex model calibration and validation can be quite costly as well. Therefore, it's important for an organization to assess the benefits to be gained and perform a cost versus. benefits analysis prior to making an investment in parametric tools.

Practitioners believe that in most cases the investment is worthwhile because the expected payoff is usually large enough. However, each organization must consider all the potential applications, and then make the decision themselves.

8.4 General Applications

Parametric techniques are used for a variety of general applications, as shown in Figure 8.1. There are other possible applications. The number is limited only by the imagination of the user.

Forward Pricing Rate Models	Subcontractor Price or Cost Analysis
Cost as an Independent Variable (CAIV)	Risk Analysis
Bid/No Bid Analysis	Conceptual Estimating
Independent Cost Estimates (ICEs)	Design-to-Cost (DTC)
Life Cycle Cost Estimates	Budget Planning Analysis
Proposal Evaluation	Should Cost Studies
Estimates at Completion (EACs)	Costing by Phase of Contract
Trade Studies	Sensitivity Analysis
Basis of Estimates (BOE's)	Affordability
Cost Spreading	Cost Realism
Sizing parameters	MTBF, MTTRs
Make-buy analysis	

Figure 8.1 General Parametric Applications

Most of these applications listed in Figure 8.1 are widely used throughout Industry and the Government and guidance on their implementation is available.

The following sections provide some general descriptions and examples relating to most of the general applications listed in Figure 8.1.

8.4.1 Forward Pricing Rate Models

While there are many estimating approaches for forecasting indirect expense rates, the most traditional approach is known as "bottoms-up." The bottoms-up approach is generally based on detailed, departmental budget data. This process of preparing and evaluating forward pricing rates can be time and cost intensive. More recently, contractors have been implementing proprietary models to forecast these rates.

8.4.1.1 Example

Using an IPT approach, one contractor developed a model that uses forecasted sales and historical cost estimating relationships (CERs) to develop forward pricing rates. This forward pricing rate model uses forecasted sales and historical CERs to develop indirect rates. The process used to develop the forward pricing indirect expense rates involves:

- Sales forecast. The sales forecast is the major cost driver for this model. Therefore, it must be developed first. An accurate sales forecast is critical because it is the baseline from which all other costs are generated. A sales forecast should be developed using the most current plan from a contractor's various budgeting processes (e.g., business plan, long range plan, current forecast, and discrete inputs from the operating units).
- Total cost input (TCI): TCI is a commonly used method for allocating G&A expenses to final cost objectives and is further described in the Cost Accounting Standard 410. TCI is calculated as a percentage of sales. A TCI/Sales ratio would be developed based on historical experience, with the result adjusted for any anomalies that may exist.
- Direct labor/materials base. Direct labor and direct material bases are developed as a percentage of TCI. These would also be based on historical trends, adjusted to reflect any significant changes that may have occurred, or are expected to occur, in the near future.
- Labor/materials/G&A expense pool costs. Labor, material, and G&A expense pool costs are developed as a percentage of their respective bases. When estimating pool costs, the model should utilize historical trends that separately consider fixed, variable, and semi-variable costs.
- Fixed pool costs remain constant from the prior year and are adjusted for any known changes (e.g., purchase of a new asset that would result in a significant increase in depreciation) and escalation.
- Variable pool costs are calculated by applying the variable pool cost portion of the rate (based on historical trends) to the current forecasted base. An example of an adjustment for a known change to variable costs

would be an accounting change where an indirect labor category is changed to a direct labor category.

- Semi-variable costs are also calculated and analyzed separately. An example of a semi-variable cost would be payroll taxes. Payroll taxes are calculated as a percentage of salary, up to a certain threshold, and fixed costs after the threshold is reached.

A projected pool may need to be adjusted for large dollar nonrecurring items such as environmental clean-up costs. Forecasted expenses would also need to be adjusted to reflect implementation of new processes. For example, if a contractor implemented a new quality system, its costs should be reflected in the forecasted expenses. The fixed, variable, and semi-variable costs would be added to arrive at total pool costs.

8.4.1.2 Evaluation Criteria

These elements should be considered when developing forward pricing rates using historical CERs:

- Reasonableness of the sales forecast;
- Accuracy of the historical CERs;
- Accuracy of the underlying data used to develop the historical relationships.

As stated, the reasonableness of the proposed sales forecast is critical because it forms the basis for the forecasted indirect rates. The forecasted sales data should be based on known (firm) sales and projected sales adjusted for probability factors (i.e., probability of obtaining a sale). This sales data should accurately reflect those amounts reported in the contractor's current budget data.

The contractor should also ensure that the historical trends developed are accurate and that they form a reliable basis for predicting future costs. The contractor should perform statistical or another form of analysis to validate that a strong relationship does exist between the two variables (e.g., sales as the independent variable and TCI as the dependent variable). The relationship should also be tested on the normalized data (cost history adjusted for any anomalies that may exist). Risk analysis may be performed to identify the cost elements that generate the most risk (i.e., have the most significant effect on the model's outputs), and which may require additional validation or adjustment. Risk analysis is discussed in Section 8.4.4, Risk Analysis. Regardless of the type of analysis performed, the effects of fixed, variable, and semi-variable costs should be considered.

The accuracy of the historical inputs used to develop the estimating relationships should also be examined by reviewing the historical pool and base costs, as well as any adjustments and anomalies. The results of previously incurred cost audits and other reviews may have an impact on the historical trends developed.

8.4.1.3 Benefits for using a Parametric Application

The proper use of a forward pricing rate model should facilitate the negotiation of a forward pricing rate agreement (FPRA). An IPT consisting of representatives from the company, DCMA, and DCAA may find that the implementation of a forward pricing rate model can save both the contractor and Government significant resources, including reduced costs and cycle time related to proposal preparation and review. When properly implemented, forward pricing rate models will be as accurate as other estimating approaches.

8.4.2 Subcontractor Price or Cost Analysis Using Vendor Data

FAR 15.404-3, Subcontract Pricing Considerations, defines the requirements for performing price or cost analysis on applicable subcontract estimates to establish the reasonableness of proposed prices. Chapter 7, Government Compliance, discusses this criterion. A variety of parametric techniques can be used to develop independent estimates for use in performing such price or cost analyses.

8.4.2.1 Data Collection, Calibration, Validation and Estimating Techniques

Developing parametric subcontractor analysis tools is similar to building tools to estimate an internal effort. The model development life cycle is identical, consisting of data collection, calibration, validation, and establishing estimating procedures that describe the methodologies for performing subcontract price or cost analysis using parametric techniques. The only substantial process difference relates to the collection of subcontractor technical, programmatic, and cost data. The following is a series of questions and answers that takes readers from a theoretical view of the modeling process to a step by step “how-to” example on how this technique can be used to evaluate the reasonableness of subcontractor estimates.

Where do you start?

The starting place for data collection is the subcontracting organization’s enterprise resource planning (ERP) system or other management information system (MIS). A user should be able to access a database containing information such as purchase order history, and from this collect part number, quantity, pricing, and schedule information. Weight data for major subcontracted items should be available from an organization’s technical group responsible for weight determinations. In general, detailed data such as cost improvement curve slopes and part counts should be obtained directly from the subcontractor via request for information or through on-site fact finding. Independent subcontractor price or cost analysis may also require broadening the data search to include public domain data sources to fill in the blanks.

How much data do you need?

When obtaining data for performing independent analysis, the goal is to seek as many relevant data points as possible. In many cases, patterns or trends will be apparent in the multiple data points and help in their normalization.

What do you have once you collect this data?

Often, regardless of the source, the results can be a large spread of data points that require normalization analysis. See the discussion about normalization in Chapter 2.

Are the data in a useable condition?

The basic purchase order data raises a number of questions related to the data (the same issues apply to in-house data). Figure 8.2 lists some typical data details that need to be assessed as part of the data collection process.

Identifying the data	• What year dollars are represented? • Does the difference between the delivery date and purchase order date represent the manufacturing span time? • Does this data include G&A and profit? • Can the delivery date be used to derive the economic base year? • Was this item purchased as a spare, a production item, a test part, or a repair item? • Does this item include engineering changes?
Evaluating the data	• Economic base year. • Manufacturing span times/schedules. • Buy quantities. • Cost make up. • What's the product source: domestic, foreign, or co-produced? • Are any prices based on option agreements? • Were any purchase orders combined with another procurement? • Did any vendors change within the collected data?
Normalizing the data	• Converting data to a constant economic year dollar value. • Determining span times/schedules and using the results for cost modeling. • Eliminating extraneous data and anomalies such as foreign/co-producers. • Did the data separate the recurring and non-recurring effort? • Deriving slopes, intercepts, midpoints, and so forth. • Who performed testing on the subcontracted items?

Figure 8.2 Data Collection Details

There is no single, convenient source of information that facilitates the normalization process, so the search may need to cover other sources, such as old estimates, proposals, contract change notices, estimating library, desks, file cabinets, company archives, and memoranda of negotiation. A memorandum of negotiation often proves to be a valuable data source because it identifies the basis for the negotiated unit price, and it provides options for adjusting this price based on quantity and rate per month changes. Chapter 2 discusses data collection and analysis in further detail.

The graph show in Figure 8.3 shows how an estimator may project part cost from the normalized data.

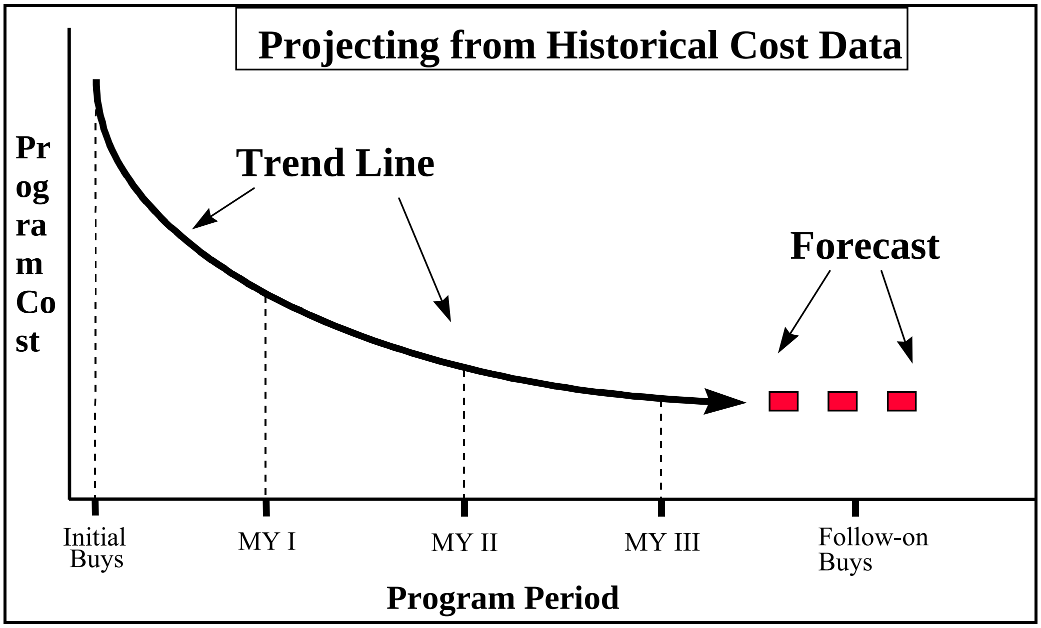


Figure 8.3 Projecting from Historical Data

The data set used in this example exhibited extremely stable properties and produced an excellent modeling result. Do not expect all results to achieve this close a fit.

8.4.2.2 Cost Analysis in an Evolving Technical Environment Using Complex Models

Are calibrated complex models suitable for analyzing analogous subcontracted subsystems?

Generally, without adjustment, the answer is "No." However, one of the greatest strengths of parametric analysis is the inherent flexibility of today's complex models. Once a model has been calibrated to a known technology or process baseline, it can be objectively modified to account for most variations. As an example, a common scenario for one contractor would be the transition from aluminum to advanced composite structures. Assume that an F-16 aluminum landing gear door subcontractor is preparing a quote to produce carbon fiber doors for an advanced fighter application. The prime contractor has calibrated its complex model for F-16 landing gear doors.

How can the contractor use its model to develop a "should cost" value for the new product?

The calibration data tells us:

- Materials used in production;
- Weight of the item;
- Parts used in assembly;
- Production quantities;
- Learning curve slopes.

With this information and basic physical parameters (e.g., approximate weight, dimensions, material mix) the prime contractor can adjust its existing model. The most significant change would relate to the material composition of the doors. In general, the switch to composite materials will ripple through weight, parts count, learning curve slope, and production process. Most complex models are able to adjust for the effect of such changes. This basic modeling technique allows the prime contractor to develop a should cost target for an advanced technology subsystem, while maintaining an audit trail back to the calibrated historical base line. Similar techniques can be used with most complex parametric estimating models.

8.4.2.3 Benefits for Using a Parametric Application

The most significant benefit for using parametric tools to perform subcontract price and cost analysis is flexibility and repeatability. Once the foundation model has been created, subsequent modeling exercises for updates and so forth can be performed with almost minimal effort. The implications for cost savings are obvious.

8.4.3 Cost As an Independent Variable (CAIV)

In March 1996, Department of Defense (DoD) Directive 5000.1 stated that, "Cost must be viewed as an independent variable. Acquisition managers shall establish aggressive but realistic objectives for all programs and follow through by trading off performance and schedule, beginning early in the program."

CAIV is an acquisition strategy that helps maintain cost objectives (including life cycle costs), while achieving the necessary performance objectives of a contract. DoD Directive 5000.2-R defines CAIV as "an acquisition philosophy put forth as policy that integrates proven successful practices with new promising DoD initiatives, to obtain superior yet reasonably priced warfighting capability."

The basic concept of CAIV is that each acquisition program has three significant variables: performance that satisfies operational requirements, affordable life cycle costs, and delivery according to an established schedule. Under the CAIV philosophy, performance and schedule are considered dependent on the funds available for a specific program. The purpose of CAIV is to reduce life cycle costs; reduce program development and production time; provide for innovative design in manufacturing, support, and contracting approaches; consider life cycle costs; and be flexible and able to overcome program cost growth. Parametric models are key tools in performing CAIV analyses. This chapter provides a brief overview of CAIV; see Appendix D for a list of additional resources.

8.4.3.1 CAIV Using Parametrics

Defense contractors must produce cost effective and performance-driven products. With this concept in mind, contractors should integrate cost estimates with program performance evaluations, and also include a cost analysis with their trade studies. To effectively perform trade studies and efficiently consider cost in each case, programs that require CAIV must build a cost model that baselines the program, estimates total ownership cost (TOC) for every trade study alternative or option, and tracks cost against targets and goals. TOC attempts to capture a product's life cycle cost, including any required peripheral support equipment and services necessary for making full use of the asset. This working cost model is a parametric model.

After an appropriate parametric model has been obtained and calibrated, a program cost and performance baseline is created. The purpose of the baseline is to establish a set of program cost and performance concepts that reflect the initial program configuration(s). This program baseline should allow trade studies to be performed using a consistent set of estimating parameters, guidelines, and assumptions.

After the program baseline has been established and reviewed by the customer, effective trade studies can be performed against it. The important aspect of trade studies is to determine the cost/performance/value of the various trade-off options (that is, the evaluated "delta" cost and performance among the options). As a result of this process, program management can assess the "best value" among the trade alternatives. "Best value" is defined as the option that meets program performance objectives at an affordable (generally lowest) cost. In some cases, performance may be traded for cost.

8.4.3.2 Implementation

The CAIV process is highly analytical and technical. It requires skilled personnel, sophisticated analytic tools, and specific institutional structures to handle the technical studies required, to track progress, and to make reports capable of initiating action. In addition, the CAIV concept must be built into the contract structure so that all parties to the program are properly motivated towards a "best value" objective. Key tasks in the CAIV process are:

- Target setting and subtask allocation;
- Contract and subcontract definition and incentives;
- Technical analysis;
- Cost progress calculation and tracking;

- Cost progress and trade study log;
- Reports.

Detailed implementation guidance on these tasks is outside the scope of this Handbook. Additional information is available on the DoD Acquisition Deskbook and CAIV web sites listed in Appendix D.

8.4.3.3 Benefits for Using a Parametric Application

The parametric model serves to help routinely evaluate prevalent (and evolving) cost estimates against cost goals and targets. Using the cost model as derived within the context of the program WBS, cost targets are "flowed down" to IPTs and subcontractors. Corrective management action (i.e., additional trade studies) is taken when cost estimates deviate from IPT and subcontractor targets. Corrective action should also be taken when the current system estimate deviates from the system-level goal. Using this approach, the program should remain affordable, and TOC carefully managed. As with the subcontractor cost analysis application, once the foundation model has been created, subsequent modeling exercises for updates and so forth can be performed with almost minimal effort. The implications for cost savings are obvious.

8.4.4 Risk Analysis

Risk analysis is another important aspect of the acquisition strategy of most major programs. The consideration of different, yet possible, program events and outcomes should lead to a more realistic estimate, in spite of uncertainties associated with cost models, variability of cost driver metrics, unplanned or unexpected events, and other factors beyond control of the IPT. By their nature, all cost estimates have some uncertainty. The number of uncertainties and the associated cost impact is usually higher early in a program's development. As the program matures, uncertainties generally decrease as a result of greater design definition and production experience.

Risk analysis provides an orderly and disciplined procedure for evaluating these uncertainties, so a more realistic cost estimate can be made. Risk analysis can be performed using a number of techniques, including parametrics. Capturing program uncertainty in the variance measures of parametric estimates allows the analyst to mathematically model and quantify the risk. This method is one technique for providing financial insight into the technical complications behind the cost growth witnessed in many programs. Risk analysis provides additional information and insights to a program's decision makers.

Risk analysis is a process that uses qualitative and quantitative techniques for analyzing, quantifying, and reducing uncertainty associated with cost or performance goals. IPTs are generally responsible for evaluating areas of uncertainty in the evolution of design and process development. The preferred common denominator for measuring these uncertainties is dollars. Therefore, most risk analyses are conducted as part of the many cost analyses performed on a typical program, including:

- Cost goal allocation;
- Baseline estimate updating;
- Cost trade studies.

Although program risk should decrease with time, the risk analysis process is iterative. On most programs, the risk analysis process is not viewed as a one-time, "check the box" activity. It is an on-going management activity that continues throughout the life of a program.

These cost risk analysis objectives fully support those of the program because it:

- Identifies program level confidence for the cost of each phase as a function of cost impacts from technical, schedule, and cost estimating risks, which are evaluated at a WBS level, consistent with decision making and the availability of appropriate data.
- Identifies program level confidence in the schedule of the development phases.
- Provides credibility to the target estimates and BOE (which should approximately equal the calculated “most likely” cost).
- Identifies technical, schedule, and cost estimating risk drivers for use in risk management exercises.
- Enables the current BOEs to reflect the cost and effectiveness of the planned risk handling strategies.
- Depicts how funding levels impact total program and phase specific confidence levels, assuming constant program execution plans.

The cost-risk analysis process begins with program definition and ends with management review(s). For a risk analysis to be effective, program definition must be at least one WBS level deeper than that at which the cost risk analysis is performed. For example, to support a cost-risk analysis conducted at WBS level three, program definition is required at WBS level four. This allows the analyst to capture all reasonable risks, and provides the visibility needed to eliminate overlapping and gaps in the analysis. Most commercially available parametric models possess a risk analysis capability.

The major benefit of the use of parametric tools in the risk analysis process is the fact that the tools are repeatable in this highly iterative procedure. Many “what-if” exercises must be performed during a program’s life cycle. The use of parametric tools is the only practical way to perform these exercises. See Appendix H for the Space Systems Cost Analysis Group (SSCAG) discussion of risk.

8.4.5 Bid/No Bid Analysis

Many companies bid on far more RFPs than they should. The question of whether to bid a project (or not) is a strategic one. The decision involves much more than making a profit. For instance, a company may be willing to take a loss on a project if there is significant profitable follow-on potential. Or, as another example, they may want to strategically place a product in the marketplace. Also, they may wish to get into a new business arena. There are many reasons to bid on a project. "No-bidding" a project may not be an option. Careful consideration given at the bid/no-bid stage can have an enormous impact on an organization's bottom line. Parametric tools are very useful at this stage of the business process, and can help the decision maker.

8.4.5.1 Example

Let's assume that a company is considering proposing to an RFP. The product is well-defined, but it's a new state of the art. Clearly, a cost estimate is a major consideration. How should the cost estimate be performed? Since the product is defined, parametric tools come to mind. Clearly, performing a bottoms-up estimate at this stage of the program doesn't seem reasonable. No bill of materials exists. Some type of "top down" estimate appears to be the way to go. Since at least a preliminary engineering evaluation must be done, a parametric approach should be adopted.

8.4.5.2 Evaluation Criteria

Once the estimate is complete, management can decide if the project is worth pursuing. If the cost estimate is within a specified competitive range, a "bid" decision could be made. It is important to note that other criteria besides cost are important considerations. Obviously, program technical competence is also important. On the other hand, cost can be a "show stopper." If the cost estimate is too far outside the competitive range, or too much money needs to be invested, a "no bid" decision would be made.

8.4.5.3 Benefits of Using a Parametric Approach

Bottoms-up estimating approaches are not practical without a BOM. Parametric models are also easily "tweaked" for subtle (or not so subtle) changes to specs, design, schedule and so forth. The benefits of parametric models are clear whenever rapid estimating turnaround is required.

8.4.6 Conceptual Estimating

Parametric costing models are powerful tools in the hands of management. If a software or hardware product has been conceptualized, parametric models can provide a fast and easy cost estimate. An estimate in the conceptual stage of a program can be invaluable in management's planning process. Engineering concepts such as weight, manufacturing and engineering complexities, source lines of code and so forth are normally available at the time concepts are being developed. Given this fact, use of parametric costing tools is the only reasonable way to perform a cost estimate this early in a program life cycle.

8.4.6.1 Example

For example, if a Government program manager has a new technical need, let's assume for a specific software application, and that application can be conceptualized, a cost estimate can be obtained. The conceptualization will take the form of a general description: application, platform, programming language(s), security level, programmer experience, schedule, an estimate of the SLOC and so forth. Such information will provide a cost estimate using any number of parametric models. A cost estimate is important early in a program, if only to benchmark a proposed technical approach.

8.4.6.2 Benefits of Using a Parametric Approach

Parametric tools were designed for just these types of applications. Parametric model inputs are largely conceptual. If a new program is being considered by the Government, concepts such as performance parameters are what come to mind. Parametric tools are utilized, with a good deal of management input. There is no substitute for experience when an estimate is being created, regardless of the technique used in the estimating process.

8.4.7 Independent Cost estimates (ICEs)

Sometimes, it's important to have an independent view of a cost estimate as performed by another technique. The key word is "independent." Such estimates are called "independent cost estimates," or ICEs. Parametric tools are invaluable for performing ICEs, that is unless the parametric tools were used to generate the primary estimate. ICEs can also be used to validate EACs for ongoing contracts.

8.4.7.1 Example

Assume, for a moment, that a bottoms-up estimate has been (or is being) generated for an organization who is proposing on a "must win" program. Winning the program could be important for a variety of reasons. In any event, should the organization bet winning the program on just one estimating approach?

If an independent estimate (or two) is performed by a team of people who have no vested interest in, or little knowledge of, the primary estimate, the benefits of such an estimate could be enormous. If the ICE supports the primary estimate, then added confidence is automatically placed on the original estimate.

If parametric tools are utilized for the ICE, many unique estimating criteria (primarily technical criteria) are used for the cost evaluation. If, on the other hand, the two estimates indicate significant differences between each other, an evaluation can still be performed to correct the primary proposal. For instance, if two estimates show a more than 10% difference (management will determine the level of significance) between themselves, a careful review, analysis and explanation may be in order.

In a "real world" example, when the two estimates differed by a significant amount, an analysis revealed that the estimating team had duplicated estimates in various places within a complex WBS. Without the ICE, the discrepancy never would have been discovered.

8.4.7.2 The Benefits of Using a Parametric Approach

If a bottoms-up estimate was used as the primary estimate, and a "second opinion" is required, the use of parametric tools is certainly an option. Other approaches (e.g., the Delphi technique) could also be used. A parametric model will link the technical parameters to the bottoms-up cost estimate. This link is invaluable.

8.4.8 Design to Cost (DTC)

Design to cost principles have been already covered within the context of CAIV, so little will be said about the technique in this section. There are subtle differences, however. CAIV is a management approach that emphasizes cost at the program level. DTC's focus is on the manufacturing aspect of a program. The term "design for manufacturability" could also be used. In any event, the approaches are similar in philosophy. Trade studies play a significant role in both approaches.

8.4.8.1 Example

Design to cost techniques are used extensively in commercial manufacturing. It stands to reason, then, that many DoD applications would exist. A customer will not purchase a $10,000 vacuum cleaner, regardless of how good it is. The product must be manufactured for customer affordability. Based on analyses, cost targets (sometimes called "standards") are allocated to manufacturing operations. The management expectation is that those standards will be met or exceeded by the floor operators. T he targets are "rolled up" to the final product. If the final target is met, the product is affordable and the manufacturing processes are under control. If the targets are exceeded, the processes may be out of control, and an analysis must be performed.

8.4.8.2 The Benefits of Using a Parametric Approach

Parametric tools fit into this process as analytical techniques are used to reduce the cost of manufacture. For instance, a parametric model can easily support manufacturing process trade-offs. Two or more competing processes can be quickly modeled that allow the manufacturing engineer to select the least costly approach. No other estimating technique is quite as effective within the context of cost trade-offs.

8.4.9 Life Cycle Cost Estimates

A parametric model can be developed to assist in estimating the life cycle costs of a program and can be used in performing "what if" analysis and the impact of technical, schedule, and programmatic factors on the total costs. For example, the Air Force needed to develop a life cycle cost model that could be used to assist in estimating the costs for aircrew training on the C-17 program. Due to the significant number of parameters that can impact the training costs they developed a parametric model that would take into account the various factors (both cost and operational factors) that allowed them to estimate the total program costs for various training requirement scenarios.

8.4.9.1 Example

The C-17 Aircrew Training System (ATS) Life Cycle Cost (LCC) model was constructed around the approved Wright Patterson AFB, Aeronautical System Division WBS for Aircrew Training Systems as modified for the C-17 ATS program. Structuring the LCC model around this established WBS provided the framework so that the model is generic enough to fundamentally analyze any integrated training system. In addition, the use of this approved WBS allows the model predictions to be easily integrated and compatible with a prime contractor's accounting system used to perform cost accumulations, cost reporting, budgeting, and cost tracking.

Rather than make the C-17 ATS LCC model a pure accounting type of LCC model, a decision was made to integrate the functional parameter algorithms of each WBS element with the cost element relationship algorithms of each WBS element. This caused the model to be more of an engineering type of model in which the predicted outputs of the model are sensitive to input data, and changes in the input data to the model (i.e. program and pragmatic data changes). As a result, the model can be used to make early predictions for program development and acquisition decisions and can then be re-used during the operations and support phase to make continuing economic decisions based on actual annual operational decisions.

The C-17 ATS LCC model was developed using the Automated Cost Estimating Integrated Tool (ACEIT) modeling environment. See Appendix A for more information about ACEIT. This environment was selected primarily because of the flexibility and functional capability that it could supply to an LCC model. As a result, the model is a tool which can be continuously employed to manage the life cycle cost of an integrated training system and is not just a single point LCC estimate.

During the development of the C-17 ATS LCC model, each WBS operational, functional, and cost element relationship rationale, element algorithm and estimating methodology, along with other relevant data and information denoting how an element is modeled, was recorded and documented. At the conclusion of the modeling effort, this documentation provided the necessary traceability information and auditing data to verify and validate the model's predictive capability. This documentation was developed online as an integral part of the definition and development of the operational, functional and hardware cost element relationships for each WBS element and associated WBS element component parts.

8.4.9.2 Benefits of Using a Parametric Approach

Because of the automated capability of the model, the vast array of integrated cost analysis tools embedded within the model, and the complete online documentation features of the model, the information necessary to understand what elements within the system are cost drivers, why these cost drivers exist, and which LCC inputs have the greatest influence on these cost drivers is readily available to the analyst. The complete online documentation provides different analysts using the model the ability to easily adapt the model to their specific needs, and provides program management with a quick and flexible method of preparing required program cost and budget reports. As before, once the one-time effort of creating the basic model is finished, model reuse and repeatability for sensitivity studies is easy.

8.4.10 Budget Planning Analysis

All businesses must budget. A decision that must be made has to answer certain questions. First, what products will be offered for sale, or proposed for future sale? Next, how will the organization fulfill its obligations? What resources will be used, and how much will the use of these resources cost? What is the best mix of business that will allow the organization to maximize profit potential? All such questions involve trade-offs. Parametric tools are useful in performing budgeting exercises. The benefits of using parametric tools for trade-off analyses have already been discussed.

8.4.11 Proposal Evaluation, Red Team Reviews

When a proposal is submitted to a customer, how good is it? What will the customer think of it? Red Team Reviews are performed by independent, senior organization personnel who place themselves "in the shoes of" the customer, so to speak. Red Team Reviews take place just prior to proposal submission. The reviews are about much more than just cost – technical compliance is often at least as important – but cost, especially if the program has a CAIV requirement, is always important. Any proposal evaluation that includes a sound parametric approach that makes a solid link between technical design and cost, will always be a superior product to the simple cost estimate, and better received by the customer. The Government also uses teams and parametric approaches to evaluate contractor estimates and proposals.

8.4.12 Should Cost Studies

Many cost evaluation techniques can be used to perform should cost studies. Selection of a technique can depend on various factors such as timing, available resources, cost of using a specific approach and so forth. Should cost studies can be performed at any time during a product's life cycle, even after a program is over. The only criterion is if management needs or wants to know the answer to the question, "What should the product (have) cost?" Depending upon the specific situation, parametric tools may or may not play a role in a should cost analysis.

For example, if a parametric model has already been developed and utilized for a program or product, the model can be used in a should cost analysis. If a program has overrun, for instance, the model can be used to evaluate differences between the proposed program and the one that was executed to determine why. In this type of application, the benefits of the parametric modeling technique should be easy, quick and efficient.

8.4.13 Estimate at Completion (EACs)

Estimates at completion are often performed on larger programs, especially if there is an earned value analysis requirement. Even so, they are normally performed as an indication of the financial health of a program. If a parametric model has been developed for a program, the model is an easy and efficient way to develop EACs on a regular basis.

8.4.14 Costing by Phase of Contract

Costing by phase of contract comes within the context of life cycle costing (LCC) or total ownership cost (TOC). Clearly, LCCs or TOCs necessitate the costing of all program phases. Sometimes, if a LCC or TOC has not been performed, and a parametric costing model has been already developed, the cost of future phases of the program can be easily evaluated. Such a data point is often required for planning purposes.

For example, if a program is in the development phase, and a production phase cost estimate is required, a parametric model can be used for the cost estimate. Again, a bottoms-up approach is not practical without a BOM.

8.4.15 Trade Studies

Trade studies always come with a CAIV or a DTC program. But, sometimes, trade studies are important within their own right. Trade studies are almost always used in a commercial business environment, and are more and more frequently being used in DoD. Trade studies are most frequently used to evaluate cost and performance trade-offs between competing technical designs. The basic idea behind trade studies is to get the highest performance for the lowest cost. What it doesn't mean is "cheapest." I t means best performance value, or best "bang for the buck." Parametric costing models are very effectively used in trade studies. Trade studies require multiple sensitivity studies, with rapid turnaround times.

For example, an Army program wanted to evaluate the performance and cost curve for gasoline and diesel engines in a tank design. Technical parameters were input into two parametric cost models – one model for gasoline engines and the other for diesel engines. Based on a performance vs. cost analysis, the diesel engine was selected. The power of the parametric models was demonstrated when the models' inputs were easily "tweaked" almost repeatedly. No other estimating approach would have been nearly as effective.

8.4.16 Sensitivity Analysis

Parametric tools are extremely powerful when sensitivity analyses are needed. As with trade studies, the ability to easily "tweak" inputs for even small changes provides management with a strong and superior analytical benefit. When a parametric model's input(s) is changed, the resulting costing is instantaneous. The sensitivity analysis can be quickly performed many, many times until the ultimate result is obtained. No other estimating approach besides parametrics can perform this task effectively.

8.4.17 Basis of Estimates (BOE's)

The traditional method for creating a basis of estimate and submitting a proposal is to go out and perform a "bottoms-up" estimate using historical data, judgmental estimates, and obtaining detailed cost and pricing data. This is a very time consuming and expensive process that is also not very flexible when assumptions relating to quantity and scope of work change. Alternatively, through the use of parametric estimating techniques a model can be developed that is less expensive in the long run and provides greater flexibility. As demonstrated in the example below, through careful planning and coordination with all groups affected, a parametric model can be created and used for the basis of estimate in many proposals. However, in order for the parametric model to be accepted as an adequate basis of estimate within a proposal the model documentation maintained by the contractor and provided to the Government reviewer and customer must contain the following information (as applicable):

- A description of the contractual effort and/or products and services being estimated by the use of the model.
- Identification of forward pricing rates and factors used in estimate.
- A description of the commercial or in-house developed model, inputs, outputs, and calibration activities used in developing the estimate.
- Time-phasing of resources estimated (labor, materials, etc.).
- A description of all assumptions and ground rules, historical databases, and judgments used in the use of or development of the model.
- A description of the process used in validating the model, including any tests performed to verify the accuracy of the model predictions using historical data or other methodologies.

8.4.17.1 Example

Lockheed Martin Astronautics (LMA) signed up as a Reinvention Laboratory member to test the PRICE H Model as a primary Basis of Estimate (BOE) based on solid, auditable and verifiable data. In coordination with the local Defense Contract Management Agency (DCMA) and Defense Contract Audit Agency (DCAA), LMA developed an implementation plan to calibrate and validate the use of the PRICE H Model. The plan included joint training, writing a procedure for the estimating manual, calibration of relevant historical programs, and development of an approach to validate and use of the model in a test proposal requiring cost or pricing data.

LMA had been using a complex commercial parametric model to develop planning estimates on prior interplanetary space programs. Technical programmatic and cost data were collected on four contemporary spacecraft to support the calibration of the model. After DCMA and DCAA reviewed the calibrations a validation test was performed by comparing the model results for selected hardware end items with current EAC projections for an almost completed space program. One the model was calibrated and validated it was used to develop the BOE for two new interplanetary spacecraft (Mars 2001 Mission Lander and Orbiter).

8.4.17.2 Lessons Learned

The following lessons were identified during the LMA project and should be considered whenever parametric estimating techniques are used to develop a BOE:

- Requires management support;
- Need company champions;
- Constant selling is required;
- There is a sizable start up cost;
- Team training is a must;
- Culture change is required;
- The implementation is a challenging learning process;
- Obtain early IPT consensus on the model and procedures;
- The data collection tasks were the most challenging;
- Interview product integrity engineers to fully understand the technical aspects of the model;
- This processes requires lots of data analysis and reconciliations;
- You will never have all the data desired but need to determine what data are critical;
- The process required about 95 percent collection and analysis and 5 percent actual calibration time.

8.4.18 Affordability and Cost Realism

One of the largest challenges both the Government and industry has is evaluating the affordability of changes in quantities and requirements. A current use is to squeeze a major concept into a budget. The concept of cost realism requires that all key performance parameters (KPPs) be costed. This activity is even increased with the constant changes occurring in funding of Government programs.

Today, most contractor and Government organizations have affordability groups. Frequently, the quantity that is anticipated when a solicitation is issued can change significantly during the negotiation and funding process. Parametric techniques can be effectively used to address the costs variances associated with changing quantities without requiring the solicitation of new bids, and provide a level of comfort relating to the affordability of the new quantities that reduces the risk for both the Government and the contractor.

In this section we will provide an example of where this was effectively used during a source selection on a $200 million Government contract for the procurement of the Joint Tactical Information Distribution System (JTIDS). The JTIDS system is used on several different weapon systems by all four armed services and NATO. This made the determination of the future requirements very difficult to estimate.

8.4.18.1 Example of Affordability Description

On the JTIDS program, the buying command used a two prong approach. The first was requiring an all inclusive set of proposal prices covering the entire range of possibilities called "price enumeration." In the context of a source selection for uncertain quantities, a discrete probability distribution is composed of a finite set of N of quantities q, for each of M line items, each with an associated probability of occurrence $p(q)$. The expected contract costs $E(C)$ is simply the sum of the products of all possible outcomes and their associated probabilities of occurrence, summed over all line items in the contract. The following equation illustrates this mathematically.

$$E(C) = \sum_{i=1}^{M} \sum_{i=1}^{Ni} (q_{ij} C_{ij} P(q_{ij}))$$

Where

q_{ij} = j_{th} possible value of q for line item i.

c_{ij} = bid unit price (cost) when buying j units of line item i.

$p(q_{ij})$ = probability of buying J units of line item i.

M = number of different line items on contract.

N = number of possible quantities for line item i.

E(C) = expected possible quantities for line item i.

For each item, it is important to note that the sum of all possibilities must equal one.

The use of this formula allowed the bidders to conduct internal "what if" analysis to arrive at their desired bottom line.

8.4.18.2 Advantages and Disadvantages

The probability approach is best used when the quantities are uncertain, current and accurate cost data are available, large quantities are procured, and there is a high unit cost. The strengths of using this approach is that it reduces the risk on the procurement, provides visibility of prices that may be unreasonable, is easy to implement, and is flexible in the fact it can be used for different line items in a contract. The disadvantage of this method is that it is sometimes difficult to get external and internal customers to agree to using this method, and since the probability given to each expected quantity can significantly effect the outcome its is imperative that these probabilities be realistic.

8.4.19 Cost Spreading

Parametric estimating tools can be used to develop time phased profiles for the total costs on a project. A top level parametric was developed based on a database of 56 NRO, Air Force, and Navy programs and was used to estimate the time to first launch. This study included space satellite acquisition, integration, system engineering, and program management costs for incrementally funded contracts.

The process used in the study identified above involved three steps: (1) estimating time from contract award to launch: (2) developing a time-phased expenditure profile, and (3) converting cost to budget. The first step required the development of the estimate for time from contract award to launch and required an independent estimate of the schedule.

The database referred to above was used as the basis for the following top level schedule model that estimates the time to first launch:

$$\text{Duration (months)} = 17.0 + 0.87W^{.406} \quad (DL*PL)^{.136}$$

Where W is dry weight in pounds, DL is design life in months, and PL is number of discrete physical payloads such as antenna arrays, sensors, or experiment packages.

This model has a very respectable 25 percent standard error of the estimate, zero bias, and 0.69 Pearson R^2 between actual and estimated schedules. The model is based on analysis of over 150 different combinations of functional forms, independent variables, and database specifications.

The second step required a model for time-phasing expenditures. A data base of 26 NRO and Air Force expenditure profiles was assembled from contractor cost-collection systems or earned value management systems. The data was input into Rayleigh, Weibull, and Beta curve models. Based on this analysis it was determined the Weibull model provided the most statistically accurate results. They were then able to develop an expenditure profile based on the Weibul model to spread the costs over the period of contract performance.

8.5 Summary

There are many uses of parametric tools other than to develop estimates for proposals. Such uses include, but are not limited to:

- Independent estimates, including conceptual estimates;
- CAIV applications;
- EACs;
- Should cost studies;
- Design-to-cost;
- Risk analysis;
- Budget planning;

Several examples were presented in this chapter. The use of parametric tools is limited only by the end-user’s imagination.

CHAPTER 9
International Use of Parametrics

Within this chapter, the term International applies to any geographic area outside of the United States of America. The chapter contrasts USA and International application of parametric estimating in three topical areas: general applications, examples, and regulatory considerations. General applications deal with differences in the way parametric estimating is applied between the International country and the USA. Examples such as basis of estimate (BOE) are presented to illustrate differences. Regulatory considerations are presented from the perspective of an organization responding to an International country request for information (RFI)/request for proposal (RFP).

The International countries covered within this chapter include France, Germany, and the United Kingdom; a section is devoted to each. Other countries haven't responded with an input. However, during the 2007 review period there is an opportunity for additional contributions.

This chapter:

- Identifies differences in parametric practices between each International country covered and the USA. While there are very few differences in practices, those that are well known are discussed.
- Addresses regulatory issues from the perspective of the International country. Regardless of its national country, any organization responding to a business opportunity in an International country is expected to conform to the regulations of that country, including the USA. Chapter 7 discusses USA regulatory considerations for use of parametric estimating.
- Can help those with little parametric estimating experience be successful within their country. It can also help those responding to business in another country be successful with submissions to that country.

9.1 France

9.1.1 General Applications

In France, parametric cost estimating is applied in much the same way as it is within the USA. Most differences are nominal naming conventions.

9.1.2 Examples

Pending.

9.1.3 Regulatory Considerations

In 2002, the President of the Republic and the French Government restated their intention to provide France with defense forces in keeping with the nation's security interests and international ambitions. This intention is reflected in the 2003-2008 military programming law, which in a demanding economic context, provides €15 billion per year for defense investments. In 2004, the armament program management reform highlighted two objectives for the Ministry of Defense activities: a) to reinforce Government project ownership b) to help develop the defense industrial and technological base at the national and European level. The Ministry of Defense is therefore responsible for building a high-performance defense system consistent with Government priorities. To this end, it is implementing a procurement policy aimed at providing the French armed forces with the equipment they need to accomplish their tasks.

This policy is based on a principle of competitive autonomy, built around two complementary goals: optimizing the economic efficiency of investments made by the Ministry of Defense to meet armed forces' requirements and guaranteeing access to the industrial and technological capabilities on which the long-term fulfillment of these requirements depends. Economic efficiency must be among the Ministry's top priorities. Priority shall be given to market mechanisms and competitive bidding, which provide great leverage in achieving competitiveness and innovation.

9.1.3.1 Marketplace Categories

The Ministry of Defense seeks to maintain and develop an industrial and technological base from three categories of marketplace segments.

- The first category groups together equipment that can be acquired through cooperation with partner nations or allies. Equipment in this category can be procured on the European market, in particular, or manufactured through European cooperation agreements.
- The second category of equipment concerns areas involving the notion of sovereignty where the nation's vital interests are at stake. Nuclear deterrent is one such area where France intends to maintain her control on technologies and preserve her ability to design, manufacture, and support equipment at the national level.
- The third category includes equipment for which the Ministry of Defense turns to the global marketplace. This includes common equipment that can be procured from a large number of providers (mobile support equipment, camouflage systems, etc.) and a few specialized systems of small quantities acquired through existing equipment.

9.1.3.2 Competitive Autonomy Principle

The implementation of the competitive autonomy principle relies on close collaboration with the French defense industry. The Ministry has undertaken to review its partnership with companies in the defense sector to ensure that French and European industrial and technological capabilities are being developed properly and that companies in the sector remain competitive. As part of this process, the Government intends to proceed with the controlled sale of its holdings in defense companies to allow them more freedom of action and promote European consolidation.

The Ministry of Defense procurement policy and its underlying principle of competitive autonomy can be broken down according to the various stages in the life cycle of an armament program:

- Preparing for the future, which entails carrying out research, controlling technologies and preparing the industrial resources required for manufacturing future weapon systems;
- Equipping the armed forces, which means preparing, designing and manufacturing equipments;
- In-service support, which is vital to the operational readiness of the armed forces.

9.1.3.3 Competitive Bidding

Procurement methods implemented by the Ministry of Defense are based on the use of competitive bidding whenever possible and making prime contractors responsible by obtaining commitments to results. The use of new procurement and financing methods is also encouraged.

A market-based approach with competitive bidding significantly contributes to technical and economic emulation and helps to improve the service provided. It provides a suitable framework for achieving a trade-off between the need to meet the public buyer's requirements at the best price and the expectations of vendors, who are justifiably concerned with the profitability and long-term future of their company. It also has a revealing impact on the competitiveness of the defense industrial and technological base. Competition is therefore desirable and is sought within an area consistent with the required degree of autonomy, particularly within the frontiers of Europe, which is the reference area.

As the consolidation of French and European industry has reduced the number of potential vendors in some fields, the situation often arises where only one company is in a position to act as prime contractor for an armament program. In such cases, the DGA (French Armament Procurement Agency) requires the prime contractor to open contracts for subsystems and equipment to competition.

9.1.3.4 Prime Contractor Responsibility

Contractual relations with prime contractors are based on a principle of responsibility designed to guarantee a well-balanced, controlled distribution of risks. This means enabling prime contractors to take the necessary action to find the best technical solutions. In exchange, they must make commitments related to the price, manufacturing lead times, and performance levels of the entire weapon system concerned. Incentives can also be negotiated to encourage prime contractors to achieve further reductions in costs and lead times during the performance of the contract signed with the DGA.

9.1.3.5 Grouped Orders

Grouped orders give prime contractors a clearer view of future work load and allow them to organize their procurements, investments, and production more efficiently on the basis of order books committing the Government over a period of several years. This approach is to the benefit of both parties and, in return, the Ministry of Defense expects substantial price reductions and better control of obsolescence from prime contractors.

9.1.3.6 New Procurement and Financing Methods

With longstanding experience in public service delegation, the French Government wishes to broaden the scope of partnerships between the public and private sectors to take in activities closer to its traditional missions. In the field of defense, studies are under way to determine how some of the armed forces requirements can be met through private funding. These new procurement methods are liable to be implemented when comparison with conventional procurement and ownership methods shows an economic advantage. It is also necessary to ensure that the transfer to the private sector of risks associated with these operations can be properly controlled. The Ministry expects new procurement methods to offer cost and funding control advantages.

9.1.3.7 In-Service Support

The in-service support of defense equipment is crucial to operational readiness. Through-life support (TLS) is a major economic consideration. Improving equipment readiness and reducing support costs are among the Ministry's top priorities. Equipment support is considered right from the armament program preparation stage. Through-life support is procured via suitable procedures and is mainly a matter for the armed forces. In accordance with the principle of competitive autonomy, competitive bidding is carried out as broadly as possible. Use of service level agreements, where contractors for support services can be made responsible through commitments to availability are encouraged. In general, "service-type" support operations can be ordered through innovative procurement procedures.

9.1.3.8 References

- *For a Competitive Autonomy in Europe: The Defence Procurement Policy*, July 2004.

9.2 Germany

9.2.1 General Applications

In Germany, parametric cost estimating is applied in much the same way as it is within the USA. Most differences deal with naming conventions. For example, funding of defense projects in Germany require a document called *Approval for Realization*. Among the contents of this document are: implementation alternatives analyzed and their assessment; justification and description of the solution selected; time and cost plans, updated and detailed; optimization of the overall balance between performance, time and cost; detailed information on expenses during the risk reduction phase; and others which are usually associated with Exhibit 300 submissions for IT project funding approval by the US Office of Management and Budget (OMB).

9.2.2 Examples

Pending.

9.2.3 Regulatory Considerations

The Bundeswehr (Federal Defense Force) is tasked by Basic Law (Grundgesetz) with the duty of providing national defense. To be able to accomplish this mission and the associated tasks, the armed forces must be provided with the necessary capabilities by making available the equipment required.

Article 87b of the Basic Law assigns the task of satisfying the armed forces' requirements for materiel and services to the Federal Defense Administration. The contracts required for providing the necessary equipment to armed forces are awarded to industry, trade, and commerce by the designated civilian authorities of the Federal Defense Administration in compliance with the awarding regulations and directives of the Federal Government.

9.2.3.1 Customer Product Management (CPM)

CPM is the defined German procurement procedure. It specifies the Bundeswehr demand delivery into the following phases: Analysis, Risk Reduction, Introduction, and In-Service.

During the analysis phase, an existing capability gap is identified and documented in the System Capability Requirement document. This is done on the basis of conceptual considerations as well as experience form everyday operations. The proposed solution is laid down in the document(s) referred to as Final Functional Requirement.

If new products are to be realized, the so-called risk reduction phase will generally follow the analysis phase. All necessary steps need to be taken to ensure that existing risks relating to performance, time, and cost aspects are reduced in all relevant project elements. Operational efficiency and suitability shall be investigated in cooperation with the future users (e.g. by means of demonstrators). Any such investigation shall be performed under near operational conditions, if this is appropriate and feasible. The risk reduction phase is concluded by a phase document referred to as Approval for Realization.

The introduction phase contains all measures required to:

- Improve fielded products/services;
- Procure available products/services at economic conditions;
- Manufacture new products at economic conditions (early partial use, if possible);
- Make them available to the user in an operational condition.

Once the preconditions for service use or early partial use, as the case may be, have been achieved, the Approval for Service Use step will be taken. If available products are procured in an unmodified condition, appropriate measures need to be taken during the analysis phase so that the Approval for Service Use can be incorporated into the Final Functional Requirement/Approval for Realization. The introduction phase is concluded by the phase document Final Report following the completion of all implementation activities.

All measures taken during the in-service phase are aimed at maintaining the operational capability and safe operation of the equipment at operational conditions and within the scope which is legally permissible until the time of disposal. If, in the course of the in-service phase, product improvement measures are necessary, a new development cycle shall be initiated in accordance with these provisions.

9.2.3.2 Information Technology (IT)

The Bundeswehr IT Office is responsible for providing the Bundeswehr with the best functional, modern, and cost-effective IT procedures and systems. It is thus the central service provider for the armed services and the Federal Defense Administration. The IT Office is composed of two policy divisions and three project divisions. The project divisions are divided by type of IT system and indicate the relatively wide scope of responsibility defined by the CPM.

- **Division C** handles all Command and Control networks for the Bundeswehr, to include military weapon control and civilian in a single network.

- **Division D** handles all Command Support IT requirements. The goal is to create an integrated network of sensor systems, reconnaissance and evaluation systems, weapon systems, and operational support systems.
- **Division E** handles all applications in support of logistic and administrative systems, which included data processing centers. The future Bundeswehr Enterprise Resource Planning (ERP) system and SASPF (Standard Application Software Product Family) will also be handled by Division E.

9.2.3.3 Contract Awards

When awarding contracts, the Bundeswehr must comply with contract awarding regulations. National or international awarding procedures are applied depending on the type and extent of required performance.

- **National Awarding Procedure.** The national awarding procedure is based on a Federal Budget Code and VOL/A (Conditions concerning Contracts for Supplies and Services). As a rule, the procedure calls for contracts awarded in a free and open competition.
- **International Awarding Procedure.** The international awarding procedure is based on the EC (European Community) Treaty and EC directives on coordination of awarding procedures within EU member countries. A series of EC directives applies; German procurement law had been written to make national awarding procedures compatible with these directives.
- **Types of Awards.** There are three types of awards or contracts: public competitive bidding, restricted bidding, and negotiated contracting. Each of the three types is defined by a specific procedure. One of the types (negotiated) includes a competitive dialogue and defined negotiation procedure as part of the award process.

9.2.3.4 Contract Terms

The drafting of contracts is based on the principle of freedom of contracts. There are no special legal provisions governing the contents of contracts with public customers. In accordance with the principle of self commitment of the administration, however, the procuring agencies are obliged to follow uniform administrative guidelines when contracting. To become legally effective, general contract terms must be clearly identified as contractual provisions. A contractor's general terms and conditions are not accepted.

9.2.3.5 References

- *Customer Product Management (CPM), Procedural Regulations for the Determination and Meeting of the Bundeswehr Demand*, Ref. 79-01-01, 24 May 2004, Bonn Germany.

- *The Bundeswehr as a Customer, Organization Procedures Contracting*, Ref. BWB-Z3.1, 10 April 2006.

9.3 United Kingdom

9.3.1 General Applications

In the United Kingdom, parametric cost estimating is applied the same way as it is within the USA. See Chapter 7 for a discussion on government compliance.

9.3.2 Examples

Pending.

9.3.3 Regulatory Considerations

The Acquisition Management System (AMS) written and maintained by the Defence Procurement Agency (DPA) contains guidance on how programme offices and occasionally Industry should prepare a Business Case (BC) for Investment Approvals Board (IAB) submissions. Director General (Scrutiny+Audit), issued a requirement for the use of properly calibrated and validated cost models (including) parametric estimating techniques on internal UK Ministry of Defence (MoD) BC submissions to the IAB.

This requirement is for all cost models to undergo a “validation and verification” (VNV) process. This will lead to the proper and consistent use of calibrated and validated cost estimating techniques in all UK MoD IAB submissions. Since much of the IAB Business Case programme costs are founded on Industry submissions it follows that the VnV process must apply equally to costs estimated both internally and externally.

This section highlights the elements of the UK Government and MoD procurement requirements. The section also discusses the key elements that should be included in an organisation’s parametric estimating system policies and procedures.

9.3.3.1 Single Source Procurements

The UK has few laws that impact directly upon the contracting process between a company and the Government. A Review Board for Government Contracts outlines procedures and process guidelines to provide a framework for non competitive contracts (the "Yellow Book") but this is a voluntary agreement and stops short of the statutory infrastructure that exists in countries such as the United States of America. For single source contract pricing, a pan UK MoD group, the Pricing & Forecasting Group (PFG) provides both technical and accountancy support to contracts officers and also provides a reciprocal service to overseas Governments.

UK MoD has a number of standard contract conditions (Defence Contract Conditions or "Defcons") that may be included within single source procurement. The main Defcons are Defcon 648 (Availability of Information) and Defcon 643 (Price Fixing and Equality of Information). A key element in the policy is "Equality of Information" that applies up to the point of contract price agreement. This places an obligation on both parties to disclose any information that is material to the business of agreeing a fair and reasonable contract price before a final price is agreed.

Defcon 643 is used in conjunction with Defcon 648 , this latter condition secures the right for MoD to require the contractor to submit a record of actual costs incurred during the performance of the contract, and allow access to Mod staff to verify such costs. Cost or pricing data should include all factual data that can be reasonably expected to support future cost estimates and validate costs already incurred. For parametric technique use, factual data includes historical data used to calibrate the model (or in building databases for cost estimating relationships (CERs)) such as:

- Technical data (e.g., weights, volume, speed);
- Programmatic data (e.g., project schedules);
- Cost data (e.g., labor hours, material costs, scrap rates, overhead rates, general and administrative expenses (G&A));
- Information on management decisions that could have a significant effect on costs (e.g., significant changes to management and manufacturing processes).

Cost or pricing data does not include judgmental data, but does include the data on which the judgment is based. Like all traditional estimating techniques, parametric estimates contain judgmental elements. Any judgmental elements are not subject to certification; however, they should be disclosed in a contractor's proposal since they are subject to negotiation.

9.3.3.2 Competitive Procurement

UK MoD does not apply Defcon 643 to competitive bids, 648 may be applied in case of follow on contracts that require single source price fixing and therefore access to contractor data to agree follow on prices. In competition, only data collected and clarified at the proposal stage will be available to Government cost estimators. Therefore it is important that all data and other assumptions necessary for consistent and comprehensive cost analysis to be supplied by the bidder are clearly identified and requested at the tender stage. This data must include the tools and calibration data used by a contractor to generate the tender prices. Examples of relevant cost and risk questionnaires are held by PFG.

9.3.3.3 Estimating Systems

Cost estimating systems are critical to the development of sound price proposals and cost forecasts. Sound price proposals provide for reasonable prices for both the contractor and the Government. Good practice criteria state that an adequate system should use appropriate source data, apply sound estimating techniques and good judgment, maintain a consistent approach, and adhere to defined estimating policies and procedures. The key issue here is to record and agree the data and assumptions used to generate the cost estimates; this record is normally referred to as a "Master Data and Assumption List (MDAL)" and is similar to the US CARD system.

A key estimating system policy and procedure for parametric estimating relates to the frequency of calibration (or database updates). Calibration is defined as the process of adjusting the general parameters of a commercial parametric model so it reasonably captures and predicts the cost behavior of a specific firm. Data collection activities that are necessary to support parametric estimating techniques (including calibration processes) tend to be expensive. As a result, parametric databases can not always be updated on a routine basis. Therefore, the use of cut-off dates is encouraged. When used, cut-off dates should be defined for all significant data inputs to the model, and included in a company's estimating system policies and procedures. In addition, contractors should disclose any cut-off dates in their proposal submissions. The parties should revisit the relevancy of the established dates before settling on final price agreement and seek updates in accordance with the contractor's disclosed procedures. When cut-off dates are not used, companies should have proper procedures to demonstrate that the most current and relevant data were used in developing a parametric based estimate.

To ensure data are current, accurate, and complete as of the date of final price agreement, contractors must establish practices for identifying and analyzing any significant data changes to determine if out-of-cycle updates are needed. A contractor's estimating policies and procedures should identify the circumstances when an out-of-cycle update is needed. Examples of some events that may trigger out-of-cycle updates include:

- New processes being implemented that are expected to significantly impact cost;
- An additional contract lot is completed, and there is a significant change in the unit cost;
- Contractor restructuring is in effect.

Contractor estimating system policies and procedures should include enough detail so the Government can make judgments regarding the adequacy of its parametric estimating practices. In addition to the key elements previously discussed, other items that should be addressed in a contractor's parametric estimating system policies and procedures include the following:

- Guidelines to determine when parametric estimating techniques are appropriate.

- Guidelines for collecting and normalizing data, including criteria for determining logical relationships and the significance of the relationships.
- Methodologies for calibrating and validating the parametric estimating techniques, including identifying significant model cost drivers (e.g., weights, volumes, software lines of code, complexity factors), input and output parameters, and procedures to validate the model to demonstrate accuracy.
- Guidelines for ensuring consistent application of parametric techniques.
- Procedures to ensure all relevant personnel have sufficient training, experience, and guidance to perform parametric estimating tasks in accordance with the contractor's estimating processes.
- Guidelines for performing internal reviews to assess compliance with estimating policies and procedures, including assessing the accuracy of the parametric estimates.

9.3.3.4 Subcontracts

In UK single source work the application of Defcons 643 and 648 flow down to sub-contracts placed by a Prime contractor. Subcontractors should be responsible for developing their own estimates since they have the experience in pricing the specific good or service they will be providing. Subcontractors are in the best position to include the cost impacts of new events such as reorganizations, changes in production processes, or changes in prices of key commodities.

For these reasons, it is a best practice for a prime contractor to obtain the necessary cost or pricing data directly from its subcontractors. Prime contractors can work with their subcontractors to streamline costs and cycle time associated with preparation and evaluation of cost or pricing data. This means, subcontractors can use parametric estimating techniques to develop their quotes, provided their models are adequately calibrated and validated. In addition, prime contractors can use parametric techniques to support cost or price analysis activities of subcontract costs. The use of parametric techniques to perform subcontract cost or price analysis is discussed in further detail in Chapter 8, Other Parametric Applications.

9.3.3.5 Cost Accounting Standards (CAS)

Cost Accounting Standards are the accounting standards, procedures and processes to which properly constituted costs and estimates, including calibrated and validated parametric estimating techniques, should comply. In certain circumstances, contracting officers may allow or request contractors to provide estimates at a specific level of detail to provide more insight into the negotiation process.

9.3.3.6 References

- Review Board for Government Contracts, the "Yellow Book"
- Green Book and Little Green Book
- JSP507
- GAR
- Acquisition Management System

Appendices

APPENDIX A

Model Builders – Commercial Software Descriptions

This appendix provides a brief discussion of available cost modeling products as provided by the companies that have consistently supported ISPA functions and conferences. Included are:

- SEER suite of products:
  - SEER-ProjectMiner™
  - SEER-Accuscope
  - SEER Spyglass Model
  - SEER DFM Model
  - SEER-DFM™ CAI Plug-in Model
  - SEER H Model
  - SEER IC Model
  - SEER-SEM
- Crystal Ball
- QSM Software Lifecycle Management (SLIM)
- r2 Estimator
- PRICE Systems:
  - True S
  - PRICE H and True H

The write-ups, below, are shown as provided by the model developers. No attempt was made to edit, recompile, or embellish in any manner. For more information about these products, go to the model developers' websites.

SEER-ProjectMiner™

SEER-ProjectMiner is an optional plug-in to SEER-SEM which produces regression-based estimates directly from historical data and provides an independent crosscheck to SEER-SEM parametric estimates. ProjectMiner uses statistical techniques to produce an estimate which combines user input with carefully matched data from a library of project models. ProjectMiner ships with a standard model library that is based upon a proprietary dataset assembled by Galorath Incorporated. You can also create a customized model library, using your own company's body of project knowledge.

Input

SEER-ProjectMiner operates within a SEER-SEM project; you can choose to activate or deactivate it using SEER-SEM's Estimate menu. When you activate SEER-ProjectMiner, it takes two types of SEER-SEM input: size parameters, and application and platform knowledge base selections.

Process

SEER-ProjectMiner uses a processed dataset, based on proprietary data assembled by Galorath Incorporated, which contains approximately 3000 projects of assorted types, from both the aerospace/defense and business domains. It uses your knowledge base inputs to select the appropriate model from this database, then creates an estimate using that model, plus your size inputs.

Output

SEER-ProjectMiner's output consists of a processing report, plus (optionally) changes to the SEER-SEM estimate itself.

You can select SEER-ProjectMiner's mode of operation:

Mode	Description
Crosscheck Mode	Displays the ProjectMiner estimate in the ProjectMiner report only, without making any changes to your SEER-SEM estimate.
Calibrated Mode	Actually changes your SEER-SEM estimate so that it is calibrated to the SEER-ProjectMiner estimate, allowing you to evaluate the impact of changing SEER-SEM parameters against SEER-ProjectMiner's estimate.
Replacement Mode	Replaces your SEER-SEM estimate with the SEER-ProjectMiner estimate. In all parts of SEER-SEM, the SEER-ProjectMiner estimate will be shown.

Summary

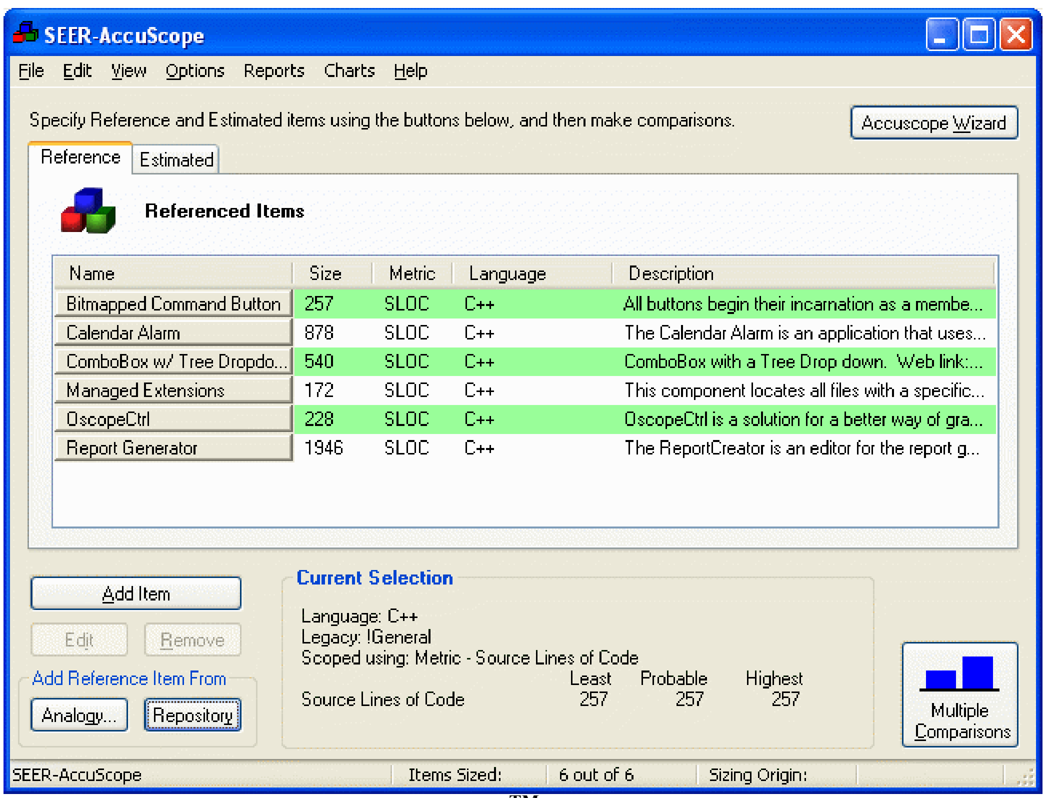
SEER-ProjectMiner allows you to check your software development project against a set of models based on a large library of historical project data, giving you an independent method of crosschecking, calibrating, and adjusting your estimate.

SEER-AccuScope™

SEER-AccuScope™ helps you size software components and other items at an early stage, when relatively little is known about them, by carrying out one-on-one comparisons between items of known size and those to be estimated. The comparisons can be subjective, intuitive "best guesses," or more precise judgments based on your knowledge of the project, and on your organization's track record in developing similar projects. You can export AccuScope's output to SEER-SEM, either as work elements or as size parameters. You can also view, save, and export reports in a variety of formats, including Word and Excel. SEER-AccuScope™'s size comparisons can be applied anywhere in the software development lifecycle; even prior to a detailed understanding of the project.

Inputs

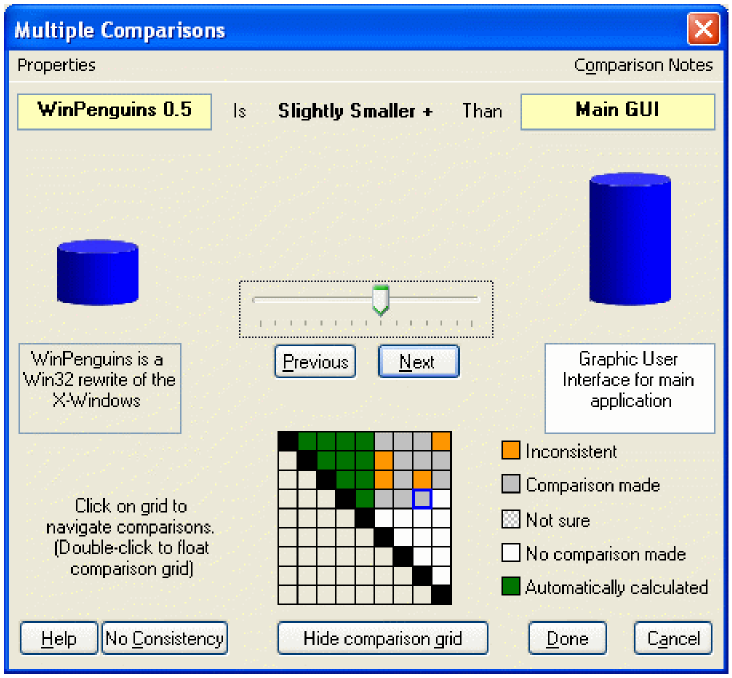
To use SEER-AccuScope™, you enter or select two basic types of input: items to be estimated, and those to be used for comparison. When you enter items to be estimated (or *estimated items*), you do not enter their size, since that's what SEER-AccuScope™ will be estimating. You do enter each item's origin (new code, existing code to be modified, integrated, etc.) and its programming language. Each comparison (or reference) item, on the other hand, does have a size. You can enter comparison items directly, in the same manner as estimated items, by entering each item's origin, language, and size (either directly, using a variety of metrics, or by analogy).



SEER-AccuScope™ Main Input Tabs

Multiple Comparisons

After you have entered all of the estimated and comparison items, you use SEER-AccuScope™'s Multiple Comparisons dialog box to make one-on-one comparisons between estimated and reference item. You make each comparison on a sliding scale, using either relative measurements (Bigger, Smaller, Slightly Smaller) or percentages. You can use a single slider to produce a single size comparison, or triple sliders for a size range. You can adjust the way that Multiple Comparisons displays its size bar graph, and set it to display a grid showing which items have been compared, as well as which comparisons are inconsistent.



Multiple Comparison Dialog Box

SEER-AccuScope™'s sophisticated comparison algorithm accounts for uncertainty in your comparisons, helping it to produce useful size estimates from incomplete or uncertain knowledge of a project. It integrates all comparisons, resulting in an improved estimate over direct comparisons alone. You can review the results of your comparisons and make new comparisons for greater accuracy.

Outputs

You can paste SEER-AccuScope™'s estimates directly into a SEER-SEM project, either as work elements (at the Program or Component level) or as size estimates for work elements. You can also save estimates as reports in Microsoft Word or Excel format. You can also export SEER-AccuScope™'s built-in reports in a variety of formats, including PDF, HTML, RTF, and plain or tab-separated text, as well as Word and Excel. Along with reports, SEER-AccuScope™ can display estimated and comparison item sizes in chart format.

Summary

SEER-AccuScope™ can serve as an extremely useful tool for estimating software size by comparison, with output that can be exported directly into a SEER-SEM project, or saved in a variety of standard formats. It uses a sophisticated algorithm which takes into account all of the comparisons in a group, as well as the inherent uncertainty of the user's input. Although its primary use is for software size estimation, it can also be used to estimate other quantities not related to software or size (e.g., length of time required for a task, projected attendance at a public event), by using the appropriate set of reference items. It should be considered for use in any situation which requires either a software-specific or general size-estimation tool.

SEER Spyglass™ Model

SEER-Spyglass™ is a "plug-in" to SEER-H™. It is used to estimate development and production costs of space-based electro-optical sensor (EOS) devices. Parametric modeling of electro-optical (EO) sensors has historically been challenging, due to a lack of data and it being a complex technical domain. The SEER-Spyglass™ plug-in expands the capabilities of SEER-H™, allowing you to perform a full lifecycle cost for space based electro-optical sensors.

The SEER-Spyglass™ specific elements and parameters are fully integrated to the SEER-H™ environment allowing users take full advantage of the SEER-H™ framework. Because of the fully integrated nature of SEER-Spyglass™, users can include system-level costs and operating and support costs. SEER-Spyglass™ specific estimation elements can be mixed and matched with standard SEER-H™ estimation elements yielding a robust estimation tool that is applicable to not just electro-optical sensors, but also to all cooperating and supporting systems. The element types specific to SEER-Spyglass™ and their definitions are listed below:

Work Element Type	Definition
Optical Telescope Assembly	An instrument used for enlarging and viewing the images of distant objects by means of refraction of incident radiant energy through lenses or reflection from concave mirrors.
Focal Plane Array	A device placed perpendicular to the optical axis of a telescope that transforms light energy into electrical signals.
Cooler	A refrigerating device designed to lower the temperature of a focal plane array and at times other parts of an EOS to make it more sensitive to low levels of radiant energy.
Mechanism	Specialized mechanical devices intended to support the operation of an EOS system.
Calibrator	Specialized devices included in an EOS system for the purpose of correcting errors that occur gradually due to aging or other causes.
Integration & Test	The set of activities required to assure that EOS subsystems have been assembled in accordance with design intent and that they are functioning within specified limits.

SEER-Spyglass™ specific Work Element types

Your selection from among the SEER-Spyglass work element types controls available technology choices from the Technology knowledge base. The costs of each SEER-Spyglass technology are largely based on your specification of the values for three to five Key Technical and Performance Parameters (KTPPs). These parameters are cost drivers that relate directly to technical features and performance capability. The KTPPs typically (but not always) differ from one technology to the next. Because costs are mostly determined in terms of KTPPs, SEER-Spyglass is well suited to trade studies based on technical features and performance.

SEER-DFM™ Model

1) SEER-DFM™ Model Overview

SEER-DFM™ provides detailed, process-based design for manufacturability parametric estimation models. Often used with SEER-H™ during the conceptual stages of design, SEER-DFM™ allows Integrated Product Teams (IPT's) to estimate manufacturing costs before committing to a design.

SEER-DFM™ applies the SEER® parametric methodology at a process level. It identifies specific cost-driving parameters for a variety of industrial processes, including machining, fabrication, casting, and PCB fabrication, and can be used for manufacturability (DFM), design for assembly (DFA), and cost avoidance programs. SEER-RateMaker™ extends this analysis to direct and indirect manufacturing cost rates at both a process specific and cross country comparison level. This section describes the modeling philosophy and methodology of SEER-DFM™ and SEER-RateMaker™.

2) SEER-DFM™ – Process based parametric model

SEER-DFM™ is a process-based manufacturing estimation model, with detailed, process-based analysis and costing for value engineering studies, and comparing manufacturing options.

As much as 80% of a product's life cycle cost may be committed during the early stages of product development. SEER-DFM™ is designed to enable engineers to make decisions that optimize the manufacturing and assembly process during the design phase, and to help them understand risk, reduce failure, and recognize opportunities to improve processes. It also helps managers understand the trade-offs and alternatives involved in aligning innovative product design with optimum manufacturing.

SEER-DFM™ can easily be assimilated into the design process to reduce downstream design changes. With SEER-DFM™ costs are not committed until later in product life cycle, resulting in savings during manufacturing and a reduction in total cost.

3) SEER-DFM™ General Concepts

The SEER-DFM™ approach to design for manufacturability analyzes cost, time-to-produce, and other baseline factors associated with product manufacture, as well as other life-cycle concerns. By modeling the many materials and processes used in production, SEER-DFM™ allows you to fine-tune manufacturing options for nearly any product, resulting in lower costs and increased efficiency, more engineering flexibility and cut "time to market".

Work Element Type	Process and Parameter Inputs		
Rollup	Covers all activities included in subordinate work elements		
Machining	Radial Mill Rough/Finish End Mill Rough/Finish Chemical Mill	End Mill Slot Angled Faces T-Sections	Reaming Tapping Hacksaw

	Shaping Rough/Finish Turning Rough/Finish Boring Rough/Finish Cylindrical Grinding Centerless Grinding Surface Grinding Rough/Finish Surface Skimming Rough/Finish Profile Rough/Finish Pocket Rough/Finish	Stringer Run Out High Speed Machining Rough/Finish Screw Machine EDM (Electric Discharge Machine) Drilling: Hand, Spade, Twist, Subland, Countersink, Carbide, Diamond, Center, Flat Bottom	Band Saw Radial Saw Broaching Automated Production Equipment Additional Items Gear Hobbing Deburr Core
Fabrication	Conventional Machines: Nibble, Notch, Shear, Punch, Brake Form CNC Turret Press CNC Laser Beam Cutting: CO2, Nd, Nd-Yag	CNC Plasma Arc Cutting CNC Gas Flame Cutting Plate Rolling Tube Bending Dedicated Tools & Dies	Progressive Dies Spin Forming Routing: Profile, Hand, Radial
Electrical Assembly	Cable Fabrication Basic Harness Fabrication	Detailed Harness Fabrication Part Preparation Direct	
Assembly	Welding: Gas Torch, Arc, Gas Metal Arc, MIG, TIG, Electron Beam, Spot	Brazing Rivet/Stake: standard, w/gasket, lubricated Bonding: Single part, multi-part, thermal	
Mold/Cast/ Forge/ Powdered Metals	Injection Molding Rotational Molding Thermoforming	Sand Casting Die Casting Investment Casting	Forging Powdered Metals Additional Items
PC Board	Blank Board Fabrication PCB Assembly	PCB Test Additional Items	
Finish and Heat Treat	Vacuum Metalizing Chromate/Phosphate Air Gun Spraying Electrostatic Coating – Fluid Electrostatic Coating – Solid	Thermal Spraying Dip Coating Brush Painting E-Coat	Electroplating Buff/Polish Additional Items Heat Treatment
Composites	Lay-Up Filament Winding	Pultrusion Composite Spray	
Detailed Composites	Detailed Part Modeling Cores Buildups Hand Lay Up (HLU) Automated Tape Lay (ATL) Resin Transfer Molding (RTM)	Resin Film Infusion (RFI) Liquid Resin Infusion (LRI) Cutting Lightning Strike Mesh Bagging / Tool Closing Hot Ply Forming	Consumables Cure Part Finish Tool Clean Tool Fabrication Non Destructive Testing
Tube, Fab, Weld & Processing	Saw Cutting Bending End Preparation	Pressure Test Weld Tubes Weld Sheet	Processing Masking Painting
Additional Items	User Specified Operation		
Purchased Part	Description Quantity	Installation Difficulty Unit Cost	
Generic Processes	Tool Clean & Prep Tool Package & Store Tool Design & Fabrication	Inspection Delays Non Destructive Testing	Mark Part Package Part

Work Element Types

SEER-DFM™ outputs not only encompass *total* process time, which includes setup, direct time, inspection and rework, but also material, vendor, tooling and other costs where appropriate. Material and labor costs may be burdened to include both direct and indirect costs.

4) Basics of Operation: Inside SEER-DFM™

Several internal mathematical algorithms drive SEER-DFM™, and describe the processes and types of assembly. Using these algorithms, SEER-DFM™ outputs direct and indirect times that may be burdened to include labor overhead, along with materials and other expenses.

When analyzing a manufacturing project, you divide its operations into work elements. The manufacturing of various basic parts might be modeled using the PC board, machining, fabrication, or mold/cast/forge process types. The integration of the resulting parts into an assembly might then be modeled as a mechanical or an electrical assembly operation.

5) General Process Type Characteristics

Following are a few basic aspects of SEER-DFM™ process types:

- **Quick work up.** SEER-DFM™ does not require extensive pre-loading of individual machine characteristics or other production rules.
- **Broad scope.** SEER-DFM™ can model a diverse set of processes. For example, Fabrication allows everything from hand craft to progressive dies.
- **Sensitive.** Assembly times are subject to quantity relations, machine tool condition, production experience, environmental restrictions, etc.
- **Design reviews.** Alerts warn you of approaches that might need changing.
- **Producibility review.** Machining and fabrication reports contain inherent trade-off capabilities. Machining: the cost of starting from raw stock versus castings. Fabrication: expenses according to the class of processes used.
- **Full time spent.** SEER-DFM™ models include direct times, set up, inspection and rework.

6) What SEER-DFM™ Gives You

SEER-DFM™'s estimates show Labor, Additional Costs, and Additional Data:

Labor	Additional Costs	Additional Data
Setup Direct Inspection Rework	Material Vendor Tooling Other	Manufacturing Index Raw Weight Finished Weight MTBF & MTTR

SEER-DFM™ Outputs

Detailed definitions of these output categories vary with work element types. Labor estimates are in terms of minutes per unit, hours per unit, cost per unit, and total labor cost. Additional Costs are per unit, total, and amortized over specified quantities. Additional data covers non-labor, non-cost items, such as manufacturing index, labor allocations, and reliability metrics.

7) Reports and Charts

SEER-DFM™ has twelve standard reports plus five charts, and users can easily develop customized reports. Report contents change interactively with input parameters, and may be viewed, printed, saved to file or copied to the clipboard for use in other programs. SEER-DFM™ reports support a variety of formats, such as Excel, HTML, PDF, and Microsoft Mail.

8) Knowledge Bases

SEER-DFM™ knowledge bases are repositories of data and information, which you can preload into a new work element as you create it. Knowledge bases supplied with SEER-DFM™ cover diverse operations, from riveted airframes to prototype electrical harnesses, to chemical milling.

9) Probability and Risk

Every SEER-DFM™ estimate predicts a range of probable future outcomes, and not an absolute estimate. Most parameters in SEER-DFM™ are entered in a Least, Likely, Most format. SEER-DFM™ uses these inputs to develop and analyze probability distributions.

10) Custom Calculations

Custom Calculations allow you to define new inputs and computation, making SEER-DFM™ completely extendible. For example:

- You are using new or nonstandard methods, for example, a new assembly process not yet a SEER-DFM™ parameter.
- You want to estimate things normally outside the scope of SEER-DFM™.
- You have an excellent understanding of your methods and your own formulas will work best. For example, a special gear grinding machine that you have kept records on for hundreds of jobs.

11) SEER-RateMaker™

SEER-RateMaker™ extends the analysis capability of SEER-DFM™ with respect to the analysis of direct and indirect manufacturing cost rates at both a process specific and cross country comparison level. SEER RateMaker™ is a powerful calculation tool that helps determine global facility costs based on individual manufacturing processes. It is designed to quickly and accurately generate labor and machine cost rates to assist in the estimating process and help to control facilities costs on a global basis.

SEER RateMaker™ provides estimators and cost analysts with a comprehensive tool, allowing them to analyze their own and supplier's manufacturing process rates for both direct and indirect labor. It enables users to perform trade studies between an in-country suppler base as well as making International cross country comparisons.

SEER-RateMaker is based on global industry standards, it contains:

- Worldwide labor cost data
- Default technology costs
- Default year and labor costs
- Costs for many regions including low cost centers
- Regional currencies
- A comprehensive range of manufacturing parameters
- An extensive list of relevant outputs

Manufacturing Rates generated in SEER RateMaker™ are automatically sent to SEER-DFM™ as required by the user. All country rates, inflation rates, and exchange rates are customizable, in addition, countries and factors can be added as required. SEER RateMaker™ while complementary to SEER-DFM™, may also be used as a stand-alone estimation tool.

12) Summary

SEER-DFM™ provides a modeling environment that can be incorporated into the design and manufacturing process of advanced technology programs. SEER-DFM™ is a process based parametric model, which provides an opportunity for engineers to assess the cost of design and manufacturing options from the early stages of design through to production. SEER RateMaker™ assists SEER-DFM™ users with the analysis of direct and indirect manufacturing rates based on specific processes, and cross country cost comparisons.

SEER-DFM™ CAI Plug-in Model

SEER-DFM™ CAI is a plug-in to SEER-DFM for providing rapid estimates for aerospace structural components and assemblies early in the design process. The modeled processes include state of practice, state of the art and emerging technologies.

Since the CAI model is a plug-in for SEER-DFM, it has the same basic user interface as the other SEER-DFM work elements. It is the work elements and knowledge bases which are unique to the CAI plug-in. CAI itself stands for the Composites Affordability Initiative, a cooperative effort between government and industry to reduce the cost of aerospace composite structures so that design engineers can take full advantage of their unique benefits. The CAI plug-in is an outgrowth of that effort, with work elements and knowledge bases specifically for estimating the cost of manufacturing aerospace composite components.

Inputs

CAI inputs consist of the following work elements, including standard SEER-DFM parameters:

Work Element Type	Description	Processes	
Fitup	Assembling detail parts or subassemblies.		
Drill Operations	All shapes, types, and sizes of fasteners.		
Composites	Individual composite models.	3D Weaving Braiding Filament Winding	Hand Layup P4A Tow Placement
Cure Methods	The method used to cure the part.	Autoclave E-Beam Fabrication	RTM VARTM
Trim	Process used to remove excess material.		
Paste Bond	Secondarily bond two materials without using an autoclave.		
Electron Beam Assembly	Fusion process for welding joints with heat from a beam of high-energy electrons.		
3D-Reinforcement	Structural enhancements.	Stitching	Z-pinning
Sheet Metal	Completing the sheet metal detail part being modeled.		
SPF/DB	Typically used to form flat sheets of titanium, aluminum, and hard metals into desired shapes using a heated die at high temperatures.		

Outputs

The CAI plug-in produces a detailed report; the actual details depend on the specific CAI work element type. The standard SEER-DFM reports and charts apply to the CAI work elements just as they do to other SEER-DFM work elements.

Security

The data embedded in the CAI Plug-in is proprietary to CAI member companies. It includes data whose export is restricted by the Arms Export Control Act and the

Export Administration Act of 1979. Access to the CAI plug-in is restricted to United States citizens and foreign nationals specifically approved by Technical Assistance Agreements as defined by ITAR.

Summary

The CAI plug-in can provide major cost savings for any organization involved in advanced composite manufacturing in the aerospace industry. Since it is fully integrated with SEER-DFM, it can seamlessly contribute to an estimate involving both advanced composite and conventional technologies.

SEER H™ Model

1) SEER H™ Model Overview

SEER-H™, part of the parametric estimation model family developed and supported by Galorath Incorporated, is a tool that aids in the estimation of hardware product cost, schedule, and risk. Estimates are generated at a system and subsystem level for product development and production, and for operations and support. SEER-H™ provides full life cycle cost modeling for advanced hardware systems. SEER-H™ also supports two plug-in models: SEER Spyglass™, a parametric model for estimating space based electro optical sensors, and SEER-IC™, which extends the Integrated Circuit estimation capability of SEER-H™ into a much more detailed, powerful parametric estimation tool. SEER-H also lets you to associatively integrate estimates for embedded software and detail production costs from Galorath's SEER-SEM (software estimation model) and SEER-DFM (design-for-manufacturability). This section describes the modeling philosophy and methodology of SEER-H™. It is followed by descriptions of the SEER Spyglass™ and SEER-IC™ models.

2) SEER-H™ Flexible Estimation Structure

SEER-H™ comprises knowledge bases that are built on real-world data and expertise. Knowledge bases can be used to form an unbiased expert opinion, particularly when specific knowledge is not yet available. As a project progresses and more design data become available, estimates can be refined as required.

Project elements are organized into a WBS, from very simple to highly complex. Six simultaneous knowledge base settings can be used. The user can audit the assessment process documenting assumptions and rationale using an on-line notes facility. SEER-H™ accepts a wide range of input parameters. Top-down and bottoms-up approaches are supported, with parameters specified using a likely, optimistic, and pessimistic approach translated into a "Least/Likely/Most" format.

3) Integrated Risk Analysis

SEER-H™ estimates a range of possible outcomes for each set of inputs. Ranges are a natural result of the uncertainty (Least/Likely/Most) specified in the inputs. The user can choose the probability level of estimates, setting a different level for each portion of the project. Work element level risk assessments are a feature of SEER-H™. "Monte Carlo sampling," Rollup and Project-level assessments of risk are also a part of a SEER-H™ estimate. Monte Carlo methods take randomized samplings and then use these to derive sample statistics.

4) Detailed Sensitivity Analysis

Users can determine the impact of adjusting specific project factors by setting references. This capability provides the information required to meet schedule and staff requirements, design-to-quality requirements, develop proposals, select contractors, and meet initial requirement specifications. Acquisition costs can be evaluated by both work element and program phase. Parameter range sensitivities are displayed in charts which illustrate the effects of parameter settings on the estimate. In addition, Rollups (summation elements) in the WBS may be excluded from the project level report outputs. This enables users to model multiple configurations within a single project, and only use one configuration with the final project estimate.

5) Knowledge Bases

SEER-H™ Knowledge Bases are information repositories of data and information describing the types of products, technologies, mission requirements, human resources, acquisition category, and O&S scenarios a user may encounter. They are designed to assist the estimator in constructing representations of the hardware, and they are updated to ensure SEER-H™ is current with other state-of-the-art models. SEER-H™ supports these knowledge base categories:

- Application
- Platform
- Operations & Support
- Acquisition Category
- Standards
- Class (Custom)

Each knowledge base pre-loads various work element parameters with appropriate settings and optional calibration information. Knowledge bases are loaded sequentially. Therefore, they may selectively overwrite other knowledge base parameter settings, depending on the order of precedence. Knowledge bases are loaded when adding or modifying a work element. The class knowledge bases are reserved for the user to customize.

6) Prediction

SEER-H™ produces gross estimates and breaks them out by activity and labor categories. Estimates are delivered via 22 standard reports and 32 charts, along with custom reports.

7) Cost (Activity) Categories

Cost categories are divided into system and subsystem for development and production, and O&S.

8) Labor (Allocations) Categories

Labor categories, or allocations, are generated for development and production, with O&S labor user defined. They are called allocations because SEER-H™ first generates estimates by cost activity, then allocates a portion of each activity into various types of labor.

9) Operations and Support Modeling

O&S modeling in SEER-H™ uses a set of parameters with the ability to create Site Work Elements. These work with SEER-H™ hardware estimates to achieve life cycle costing.

O&S estimates for hardware items are related in potentially complex ways to unique O&S support sites. SEER-H™ provides up to three support levels, each uniquely specified.

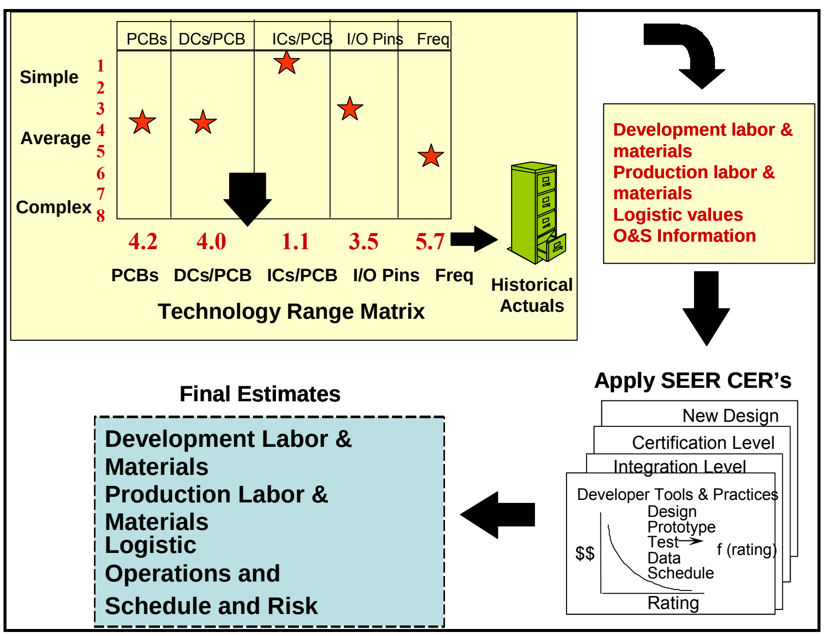
10) System Level Costing

SEER-H™ has the capability to include System Level Costs (SLC) with respect to integrating sub-systems, and systems, or systems of systems costs. System-level cost analysis can be added to your estimate by setting parameters at any roll-up level and/or for the entire project. The SLC feature uses the labor and material costs of the sub-systems along with system-level complexity factors to compute the overall system-level costs.

System-level costs are reported for both recurring and non-recurring system level activities. They are integrated into SEER-H estimates for a fully integrated system estimate.

11) Estimation Methodology

SEER-H™ uses a combination of metrics mapping and analytic techniques. Mapping databases work by analogy to assess the system against previous experience. Once analogical baseline estimates are derived, additional CERs are employed to refine the estimate for project specific parameters (see the following illustration).



SEER-H™ Estimation Methodology

SEER-H™ estimates a range of components, subsystems, and systems. SEER-H™ input parameters characterize electronic and mechanical items very differently. Technology, type, density, and circuitry quantity characterize electronics. Size, material composition, geometric complexity, and tolerance requirements characterize structures, mechanisms, and hydraulics. After metrics mapping, SEER-H™ uses several CERs to adjust costs to project specific parameters.

12) Knowledge Base and Calibration Inputs

Adjustment factors are parameters used to create knowledge bases. These factors are used to adjust any one of the final output estimates, each on a percentage basis. Three categories are available, each one having an adjustment input for each SEER-H™ output element:

- Class Adjustments
- Acquisition Adjustments
- Standard Adjustments

13) Calibration

SEER-H™ offers a full calibration capability. With calibration, a user can adjust SEER-H™ to unique hardware and organizational situations. SEER-H™ estimation accuracy generally increases as you use more of the available calibration methods described below in ascending order.

Calibration Method	Method Summary	Benefit
1. Parameter Entry with Standard Knowledge Bases	Fast and flexible workups of projects, accurate estimation out of the box.	
2. Custom Knowledge Bases	Create knowledge bases that preset the parameters in electrical or mechanical work elements.	Custom knowledge bases can be used to easily front-load known information about people, processes and hardware.
3. Custom Knowledge Bases with Calibration Factors	Enter final schedule and/or effort adjustments, obtained by comparing SEER-H™ estimates of completed projects to actual outcomes.	Estimate outcomes can be further refined, increasing accuracy. This method complies with government parametric auditing standards.
4. Custom Mapping Databases	Modify the historically based databases that are the foundation of SEER-H™ estimates.	Complete visibility into the data upon which SEER-H™ estimates are built.

SEER-H™ Calibration Methods

Calibration Mode provides the third level of calibration. Calibration Mode lets you compare *estimated* outcomes against *actual* outcomes. The difference between estimates and actuals is the *calibration factor*, with which you adjust SEER-H™ estimates by making them more accurate. Calibration adjustment factors are most typically stored in a knowledge base for convenient future use.

Calibrations are *comparisons* of actuals against estimates. Therefore, there are two sides to any calibration effort. A deficit on either side leads to a calibration that is not optimal. The more parameters and knowledge bases you specify the better the SEER-H™ estimate, and thus, the better the calibration.

14) SEER-H™ Summary

SEER-H™ provides a complete parametric life cycle cost estimation tool suite. It provides a completely flexible environment that can be customized to model advanced technologies. SEER-H™ is used for many DoD and other government sponsored projects and is widely used across industry around the world. SEER-Spyglass™ and SEER-IC™, plug-ins to SEER-H™, expand the parametric modeling framework of SEER-H™ into specialized modeling arenas.

SEER IC™ Model

SEER-IC™ is a "plug-in" that enhances SEER-H's™ ability to estimate high end integrated circuit devices (ICs), such as those typically used in the aerospace and other high technology industries. It provides a means to perform accurate full lifecycle cost estimates in an industry which has been known for its volatility.

The SEER-IC™ specific elements and parameters are fully integrated to the SEER-H™ environment allowing users take full advantage of the SEER-H™ framework to generate IC estimates. Because of the fully integrated nature of SEER-IC™, users can include system-level costs and operating and support costs. SEER-IC™ specific estimation elements can be mixed and matched with standard SEER-H™ estimation elements yielding a robust estimation tool that is applicable to not just integrated circuits, but also to cooperating and supporting systems. SEER-IC™ currently supports the following technologies:

- Application-Specific Integrated Circuits (ASICs) – Microchips that are designed for a specific purpose. Currently the most common ASIC design methodology is Standard Cell, which uses existing libraries to create physical representations of a logical design. ASICs are used in a wide range of applications, and can provide improved performance, reduced power consumption, and ultimately reduced costs.
- Field-Programmable Gate Arrays (FPGAs) – A semiconductor device that contains programmable logic components and interconnects. Compared to ASICs, FPGAs have several advantages such as shorter time to market, the ability to re-program on the field, and lower non-recurring engineering costs.

Similar to SEER-Spyglass, the costs of each SEER-IC technology are largely based on your specification of the values for three to five Key Technical and Performance Parameters (KTPPs) that relate directly to technical features and performance capability.

SEER-SEM™

SEER-SEM is a Software Estimation Model which is a part of a suite of tools offered by Galorath Incorporated that is dedicated to complete software project estimation. The suite of tools also includes a software project planning model (the SEER-SEM Client for Microsoft Project®), a regression-based model for independent comparison (SEER-ProjectMiner™), a software sizing model (SEER-AccuScope™), a repository analysis tool (SEER-ScatterPlot™), a use case sizing model (SEER-CriticalMass™), and a project monitoring tool that combines Performance Measurement and Parametric Estimation (SEER-PPMC™). Galorath also offers software suites for a Hardware Estimation Model (SEER-H™, SEER-IC™ and SEER-Spyglass™) and a Design for Manufacturability Model (SEER-DFM™ and Composites Plug-in), all of which are described elsewhere in this handbook.

SEER-SEM is a powerful decision-support tool that estimates software development and maintenance cost, labor, staffing, schedule, reliability, and risk as a function of size, technology, and any project management constraints. SEER-SEM is effective for all types of software projects, from commercial IT business applications to real-time embedded aerospace systems. This tool will help users make vital decisions about development and maintenance of software products, ensuring project plans that are realistic and defendable. The SEER-SEM Suite of tools is partly proprietary and equations are provided to licensed users. SEER-SEM is applicable to all program types, as well as most phases of the software development life cycle. More information on the SEER® family of tools can be obtained from the model vendor.

SEER-SEM Inputs

SEER-SEM inputs can be divided into three categories: Size, Knowledge Bases, and Parameters.

- • *Size:* Size can be entered in Source Lines of Code (SLOC), Function Points, Function Based Sizing, Use Cases, or user-defined Proxies. For Commercial-Off-The-Shelf (COTS) software, size can be entered three ways: Quick Size, Features (number of features used), and Object Sizing. There are separate sizing parameters for New and Pre-existing code (which is further divided into code designed for reuse and code not designed for reuse). For pre-existing code, users specify the amount of rework that will be required to integrate the existing code into the new system. This is done by specifying the amount of Redesign required, Reimplementation required, and Retest required. For each size input, the model accounts for uncertainty by using a three-point system, where users may provide Least, Likely, and Most values. The SEER-SEM Suite includes SEER-AccuScope with the SEER Repository Data Base (SEER-RDB) which are versatile tools designed for use by cost estimators and analysts, engineers and project managers.
- *Knowledge Bases*: A knowledge base is a set of parameter values based on real, completed estimates that provide industry average ranges for initial analysis and benchmarking. SEER-SEM contains knowledge bases for different types of software. Users must specify the knowledge bases to be used by the model and may add their own knowledge bases as well. The following are samples of the knowledge bases included with SEER-SEM:
  - *Platform:* The primary mission or operating environment of the software under estimation (e.g., Avionics, Ground-Based, Manned Space, or Server Based).
  - *Application*: The overall software function of the software (e.g., Command and Control, Mission Planning, Database, or Testing).

- *Acquisition Method*: The method by which the software is acquired (e.g., New Development, Language Conversion, Modification, or Re-engineering).
- *Development Method*: The method used for development (e.g., Waterfall, Evolutionary, Code Generation, or Spiral).
- *Development Standard*: The standard used in development and the degree of tailoring (e.g., MIL-STD-498 Weapons, ANSI J-STD-016 Full, IEEE/EIA 12207, or Commercial).
- *Class:* This input is primarily for user-defined knowledge bases.
- *COTS Component Type*: The type of COTS program (if any), such as Class Library, Database, or Stand Alone Application.

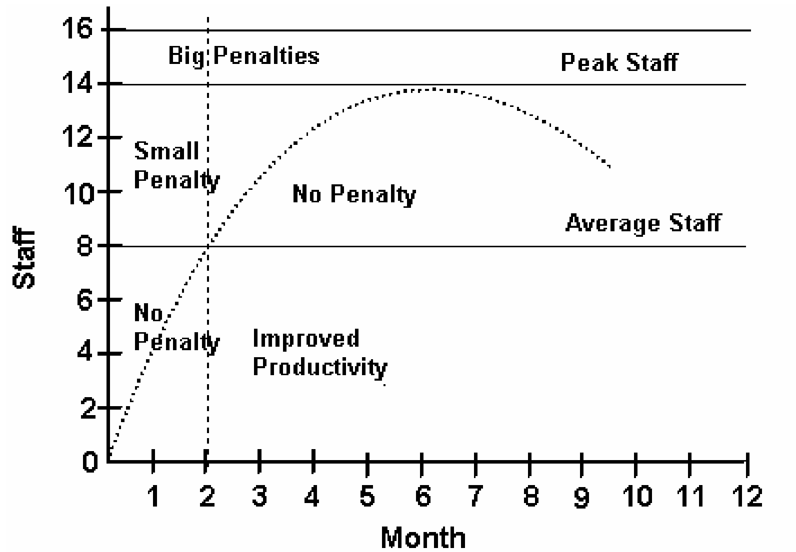
- *Parameters:* SEER-SEM contains over 30 Technology and Environment Parameters, with which users can refine their estimates. The input values generally range from "very low" to "extra high." As in size, users must specify Least, Likely, and Most values for each input. The selected knowledge base computes default values for all parameters except for size. Therefore, if users are unfamiliar with a particular parameter, they can use the knowledge base default values. The primary categories of Technology and Environment Parameters are:
  - *Personnel capability and experience*: The seven parameters in this category measure the caliber of personnel used on the project: Analyst Capabilities, Analyst's Application Experience, Programmer Capabilities, Programmer's Language Experience, Development System Experience, Target System Experience, and Practices and Methods Experience.
  - *Development support environment*: The nine parameters in this category describe the management and technical environment provided to developers: Modern Development Practices Use, Automated Tools Use, Turnaround Time, Response Time, Multiple Site Development, Resource Dedication, Resource and Support Location, Development System Volatility, and Process Volatility.
  - *Product development requirements*: The five parameters in this category are: Requirements Volatility (Change), Specification Level – Reliability, Test Level, Quality Assurance Level, and Rehost from Development to Target.
  - *Reusability requirements*: The two parameters in this category measure the degree of reuse needed for future programs and the percentage of software affected by reusability requirements: Reusability Level Required and Software Impacted by Reuse.
  - *Development environment complexity*: The four parameters in this category rate the relative complexity of the development system, compilers, file interfaces, and support environment: Language Type (complexity), Development System Complexity, Application Class Complexity, and Process Improvement.
  - *Target environment*: The seven parameters in this category focus on the target (operational system) to be delivered: Special Display Requirements, Memory Constraints, Time Constraints, Real Time Code, Target System Complexity, Target System Volatility, and Security Requirements (nominally the most sensitive input parameter in the model).

There are also parameters not related to the Technology and Environment that factor into the estimate. These include: Schedule and Staffing Constraints, Risk Inputs, Labor Rates, System Requirements Design and Software Requirements Analysis Inputs, and System Integration Inputs.

SEER-SEM Processing

Estimated effort is proportional to size raised to an entropy factor, which is usually 1.2, but can vary based on input parameters. The schedule is estimated in a similar manner, but is less sensitive to size. SEER-SEM provides a full Monte Carlo-based risk analysis, entropy adjustment appropriate to the task under analysis, staff/activity allocation within projects, and month-by-month staffing computations.

Estimated effort is spread over the estimated schedule to develop a staffing profile. The model can accommodate many different staffing profiles (including the traditional Rayleigh-Norden profile and variants, fixed staffing, mandated staffing and mixed profiles). Staffing constraints are evaluated against a project's ability to absorb staff, and productivity adjustments are made for over-staffing and under-staffing situations.



Before effort, schedule, and defects are computed, SEER-SEM makes several intermediate calculations. Effective size is computed from new and reused software. An effective technology rating (ETR) is computed using technology and environment parameters, and staffing rates are determined by the application's complexity and selected staffing profile.

SEER-SEM Outputs

SEER-SEM allows users to display output data in a variety of ways. Along with the cumulative outputs, estimates are broken down to communicate various aspects of the project. Schedule, effort, and cost are allocated among the following development activities: System Requirements Design, Software Requirements Analysis, Preliminary Design, Detailed Design, Code and Unit Test, Component Integration and Test, Program Test, and System Integration thru OT&E. Furthermore, the model provides a breakout of the labor category allocation which includes: Management, Software Requirements, Design, Code, Data Preparation, Test, Configuration Management, and Quality Assurance. Additional output options include a detailed staffing profile, cost by fiscal year, effort and cost by month, software metrics, risk analysis, a time-phased defect profile, a SEI maturity rating, and others.

The SEER-SEM Suite includes SEER-PPMC (Parametric Project Monitoring and Control), which enables you to use your software project plans for tracking and forecasting. SEER-PPMC couples traditional Performance Measurement methods with Parametric Estimating techniques providing project managers with progress tracking, status indication, performance based forecasting, and support for analyzing course correction alternatives. This capability provides indications of project health that are more timely, accurate, and closely connected to the root cause of potential trouble.

The SEER-SEM Suite also offers another option for displaying output data. SEER-ScatterPlot, a repository analysis tool that allows users to view past data, perform regressions, develop and display trends, and provide and compare them to new estimates. Users may filter datasets to the points of interest based on numerous, configurable criteria and may click on individual points to examine their values or to drop outliers. SEER-ScatterPlot also generates an equation based on the data and shows the correlation and other statistics.

SEER-SEM Support Cost Considerations

SEER-SEM contains an optional "maintenance" model that provides base year costs, effort in person months, and average staff level for each year of a user-specified schedule. Maintenance is divided into four categories: Corrective Maintenance, Adaptive Maintenance, Perfective Maintenance, and Enhancement Maintenance. Users may specify the support time period along with other support-unique parameters including: maintenance rigor (level of support), annual change rate, number of operational sites, expected size growth, differences between the development and support personnel, differences between the development and support environment, minimum and maximum staffing constraints, monthly labor rate, percent of code to be maintained, option to maintain the entire system or only the changed portion, and steady state staff level option.

Crystal Ball

Monte Carlo Simulation and Optimization with Crystal Ball® software

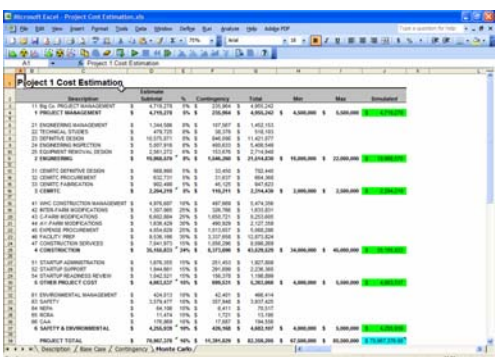
This document answers the basic questions of what is Crystal Ball, how is it used, and how is it applied in cost and risk estimation. Additional information and a free evaluation version are available on the Crystal Ball Web site (http://www.crystalball.com/ispa) and in the electronic documentation available with each version of the software.

What is Crystal Ball?

Crystal Ball Professional Edition is a suite of analytical software applications that enhance the way you use Microsoft® Excel. By introducing analytical methods such as simulation, optimization, and time-series forecasting into the spreadsheet environment, tools like Crystal Ball have raised the art of forecasting and risk analysis to a more accurate and accessible level.

What is a spreadsheet model, and does Crystal Ball require you to create new models?

We define "spreadsheet models" as Excel workbooks that represent an actual or hypothetical system or set of relationships. For example, a spreadsheet model can be an annual budget, a cost estimation worksheet, a market forecast, or even a simple mathematical function.

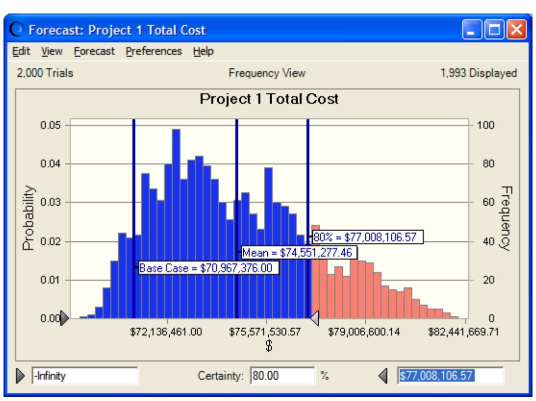


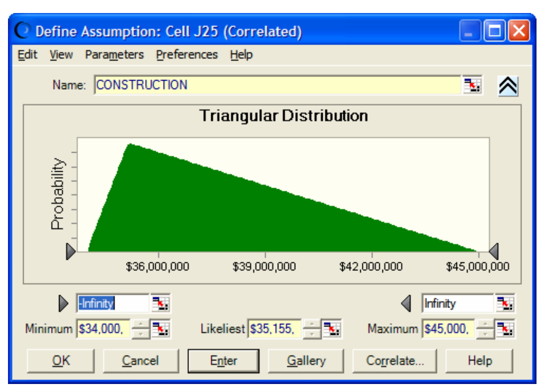
If you have already invested your time and resources to create a spreadsheet model, you can use Crystal Ball to enhance the analysis of that model, often with few or no changes to the existing format. If you are creating an entirely new model, you should first develop the model (with the expected or "base case" inputs) and once complete, start to work with Crystal Ball. Simultaneously building the model and adding simulation features often leads to an increased number of model errors.

How can Crystal Ball help your spreadsheet analyses?

At the simplest level, Crystal Ball offers two important spreadsheet enhancements. **First,** spreadsheet modelers can use Crystal Ball to convert uncertain or variable inputs (e.g., costs, returns, or time) from single values into more realistic ranges. In static, or deterministic, models, most people select an average or best-guess value for these types of inputs.

For example, rather than rely on an average (and unlikely) construction cost of $35M, Crystal Ball can replace this single point estimate with a range of values from $34M to $45M. With this method, the up side, down side, and all intermediate costs can now be accounted for within





the input variable. These ranges of values are represented by probability distributions, like the triangular distribution, that are generated from historical data or expert opinion that already exists within your company.

Second, modelers can use Crystal Ball to dynamically generate hundreds or thousands of alternative spreadsheet scenarios as compared to the single scenario of a static spreadsheet. This form of simulation, called Monte Carlo, randomly samples from the ranges defined in the uncertain inputs, recalculates the spreadsheet, stores the outputs, and repeats the process.

Simulation removes the limitations of spreadsheets by avoiding reliance on average values and providing valuable insights into the effects of variability on a forecast. Simulation also removes sampling bias (conservative or optimistic) that will occur when inputs are selected by hand rather than by an impartial number generator.

Traditional spreadsheet analysis fails to produce accurate forecasts because it is generally restricted to a limited number of "what-if" scenarios or to using the classic "best, worst, and most-likely case" approach. In both techniques, the analyst is limited to a relatively small number of alternative scenarios that provide no associated probability of occurrence. With Monte Carlo simulation, Crystal Ball simulations move you from a deterministic, or static, analysis to a probabilistic world view that recognizes and compensates for uncertainty, risk or variation.

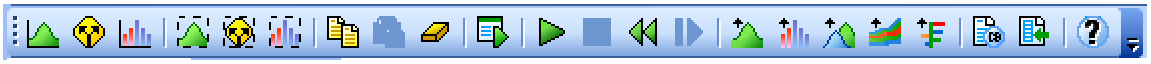
What is Monte Carlo simulation?

Monte Carlo simulation is a mathematical sampling technique that uses random numbers to measure the effects of uncertainty. One useful way of thinking about Crystal Ball is as a "scenario" or "what-if" generator. You could create several iterations of a cost estimate, or instead you could first use Crystal Ball to create thousands of scenarios through simulation, saving time.

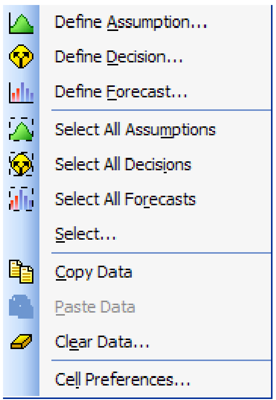
What does Crystal Ball look like in Microsoft Excel?

Crystal Ball adds a toolbar and three menus to Excel. The menus are **Define**, for setting up the simulation model, **Run**, for running the simulations and other tools, and **Analyze**, for viewing the results, creating reports, and exporting data. This pattern of define, run, and analyze represents the three primary steps required for any simulation. You may find it helpful to think of Crystal Ball modeling in this fashion. The Analyze phase usually leads back into Define as you adjust and test the model based on your continuing simulation results.

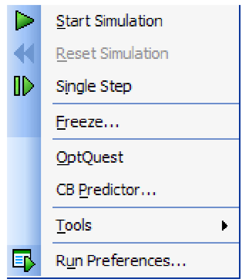
The tool bar follows this convention as well, with the Define buttons on the left, the Run buttons in the middle, and the Analyze buttons on the right.



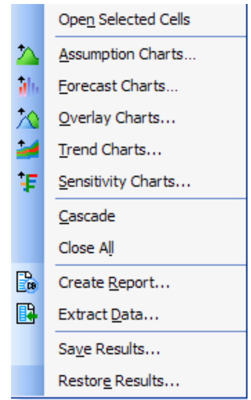
The Crystal Ball toolbar



Define Menu



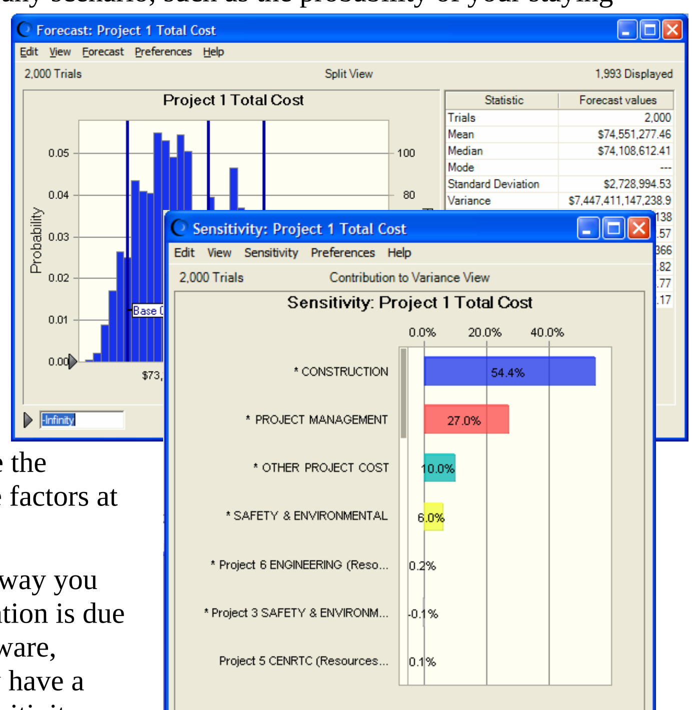
Run Menu



Analyze Menu

What does the output look like?

The primary output chart of Crystal Ball is the forecast chart. This interactive histogram displays the simulation statistics for calculations such as total cost, budget overruns, estimated resources, and others. The chart can relate the likelihood of any scenario, such as the probability of your staying within budget, and can calculate confidence intervals around expected values (e.g., 10% and 90% certainty). As of Crystal Ball version 7.2, you can view the forecast as a spilt view with multiple charts and tables.



How can you tell what is driving the variation?

The sensitivity chart is a particularly powerful method for pinpointing the drivers of uncertainty within a forecast. Generated during the simulation, this chart describes which of the uncertain factors have the greatest impact on your bottom line, with the factors at top exerting the greatest influence.

You can interpret the sensitivity chart in the way you would a Pareto Chart, since most of the variation is due to a small subset of the variable inputs. Be aware, however, that other, non-variable inputs may have a much larger impact on a specific output. Sensitivity only measures the impact of the variable or uncertain model elements.

How valuable would it be for you to know that, prior to project start, the cost of a new project will rest more on reducing the construction costs than on reducing costs for safety en environmental impact studies? Sensitivity charts can reveal the critical components within a model.

What are some benefits of Monte Carlo simulation?

Monte Carlo simulation is successfully used across multiple industries to

- Reduce the time needed to produce relevant estimates, by automating the process of producing “what if” scenarios;
- Mitigate risk in cost and schedule by giving immediate insight on variables driving the variation around the output forecast, eliminate “surprises” in cost or schedule overruns, and improve the ability to make knowledgeable decisions on where to focus resources; and
- Increase accuracy levels by replacing single point estimates with more accurate range of all possible outcomes, providing decision-makers factual data that shows the risk associated with each choice, and ensuring greater customer satisfaction.

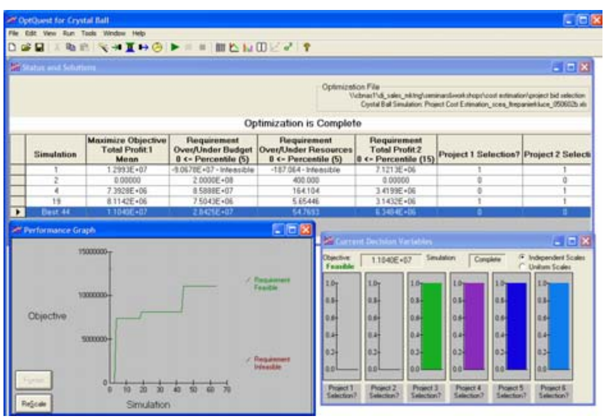
Are there example models available?

All versions of Crystal Ball, even evaluations, come with at least 30 example models that represent a variety of applications, industries, and fields. Inside Excel, you can use the Help menu to view Crystal Ball > Example Guide, which opens an Excel spreadsheet with model listings and descriptions. We offer additional example models – many of which have been contributed by customers – on our Web site (http://models.crystalball.com/).

What is optimization, and how can it work with simulation?

While simulation answers the question of “What if,” optimization answers “What’s best.” For example, simulation can be applied in cost and risk analyses to calculate the effects of variation on costs. Optimization, on the other hand, is the tool that helps to determine how to minimize costs.

In another case, if you were a project manager, you could use Crystal Ball to simulate cost and resource variations for a large number of projects. But then assume that you do not have the necessary resources to complete all the projects in your portfolio. Optimization could help you to identify which projects will give you the greatest return for your available resources, while still accounting for the uncertainty around the costs and schedules of each project.



OptQuest®, the optimization program in Crystal Ball Professional and Premium Editions, works with simulations to automatically search for and find optimal solutions.

OptQuest is a global optimizer that uses several optimization techniques such as Scatter Search and Advanced Tabu Search. OptQuest also employs heuristics, problem solving techniques that use self-education to improve performance, and both short-term and long-term adaptive memory. This multi-pronged approach to optimization is far more adept at converging on optimal solutions that single methods or genetic algorithms.

You first define the controllable inputs, known as decision variables, in your Excel model. Each decision variable is a range of acceptable values for a decision or resource.

You then apply OptQuest, which selects a set of decision values, runs a Crystal Ball simulation, examines the response, and repeats the process with another set of decision values.

What are some benefits of optimization?

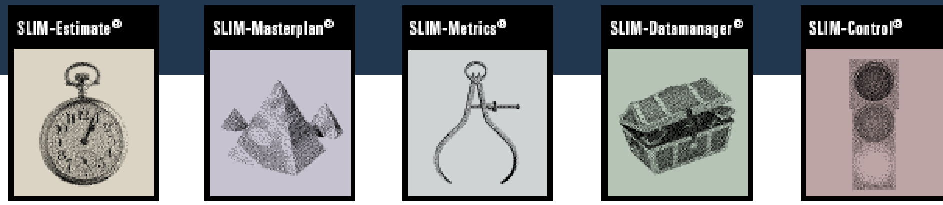
Optimization allows you to consider multiple aspects of a problem by specifying constraints, goals and requirements up front. It improves the decision time and cost, by rapidly performing searches and evaluations of possible responses, and increasing your ability to make knowledgeable decisions early in the project cycle. Finally, it provides you an optimized process, product or project, which includes variation around the inputs and allows you to simultaneously consider cost, performance and reliability all at once.

QSM Software Lifecycle Management (SLIM)

Quantitative Software Management was founded in 1978 by Lawrence H. Putnam, a world-renowned expert in the software measurement industry. Since its inception, QSM's mission has been to develop effective solutions for software estimating, project control, productivity improvement analysis, and risk mitigation. With world headquarters in Washington DC, and regional offices in Massachusetts, England, and The Netherlands, QSM has established itself as the leading total solution provider of choice for software developers in high performance mission-critical environments. Its leading comprehensive suite of products, entitled Software Lifecycle Management (SLIM) is the household brand for decision makers in Fortune 500 companies such as IBM, MOTOROLA, and EDS, as well as government and military organizations such as the U.S. Department of Defense.

World Headquarters
2000 Corporate Ridge, Suite 900
McLean, Virginia 22102
Tel: (800) 424-6755 Fax: (703) 749-3795
www.qsm.com

The Intelligence Behind Successful Software Projects



A Higher Measure of Your Success

Your mission is critical. When it comes to the success of your software project, results aren't optional.

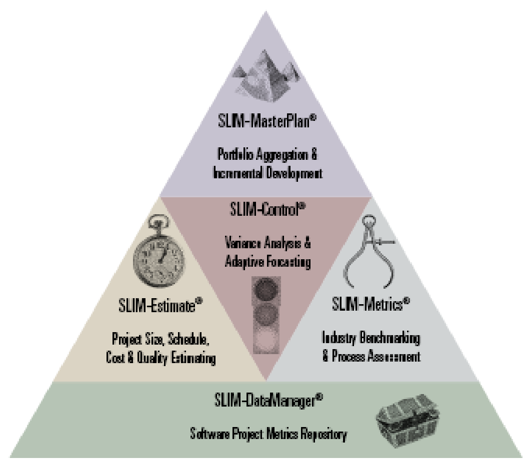
Faced with spiraling costs, limited staff, shorter deadlines, and increased competition, your success is directly linked to your ability to realistically scope your project and then communicate and negotiate acceptable schedule, cost, and reliability boundaries. Simply stated, it's your software project's inherent and learned Intelligence Quotient (IQ). With this intelligence at hand, you can reliably predict and manage success.

When reliability and results are a MUST, you need a partner who understands your mission. A partner who can help increase your project's IQ. You need Quantitative Software Management, Inc.® (QSM®). Since 1978, QSM has been helping software developers succeed by making the right choice with the lowest risks and highest rewards. We call it—the intelligence behind successful software projects.

QSM®, It's About Getting the Big Picture

QSM delivers a comprehensive solution to help you win from project planning to project completion, over the entire lifecycle of every project. QSM's core solution features a suite of Software LIfecycle Management (SLIM™) products and services that contain the horsepower you need to succeed: estimating, tracking and benchmarking.

While there are many measurement tools claiming to do parts of what you need, QSM delivers a total solution that stands above the rest in helping you get the big picture. Real value. Real fast. Consistently.



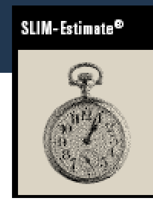
Smart Choice

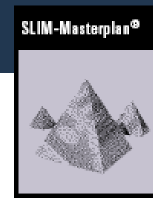
Project Intelligence

QSM®

IQ-Enabled Decisions

SLIM-Estimate®





What If Success Were Predictable?

Just imagine: your project is hailed as successful. It is completed on-time, on-budget and on-target. You get rave reviews from customers and management alike. And you can't wait for the next challenge…. To most software organizations, it could only happen in dreamland. In the real world—constantly pressured to be faster-to-market, do more-for-less and exceed customer demands—predicting success is risky business. What if there was a way to eliminate the guesswork?

Now you can stop imagining. There is a real solution to help you get there: Predictably. Effectively. Precisely. With full control and mitigated risk. It's called SLIM-Estimate from QSM, the company that delivers the intelligence behind successful software projects. Designed by people who understand your challenge, SLIM-Estimate is your "anti-guessware" arsenal to ensure requirements, budgets, timelines, staffing and product reliability withstand scrutiny BEFORE someone commits you to "mission impossible."

SLIM-Estimate helps you get there by converging your past performance with the vast experience of Fortune 1,000 companies and incorporating proven tools that are results-driven and reality-based. Bottom line: with QSM, you're not alone. You can leverage shared knowledge of 'what works' for world-class developers to predict success within your own environment.

SLIM-Estimate: The Tools You Need To Succeed

Intelligent View Technology
Process information in parallel. Customize, manipulate and analyze multiple outputs to communicate results quickly and efficiently to your customers.

Smart Component Technology
Eliminate double entry and laborious updating with dynamically linked objects that update your assumptions and solutions as you graphically operate the cockpit controls.

Quick Start Wizards
Get answers to the most frequently asked business questions quickly and efficiently with minimal input. These are ideal for early estimates when you don't have lots of information.

Automatic Solutions Generator
Enjoy the expertise of a senior consultant at your fingertips. Enter your project constraints and SLIM-Estimate will provide multiple "can-do" alternatives to solve the problem.

Tailored Estimates Based On Historic Data
Save time on customizing estimates and improve their quality with the push of a button. Just select the relevant projects and all the characteristics (efficiency, staffing, reliability, phase customization, etc.) will be transferred to a new estimate. It's instant customization.

Flexible Size Estimating With Spreadsheet Model Integration
Rapidly respond to new technology or build a custom solution for any special client need with spreadsheet sizing models which can seamlessly integrate with the five sizing methods already built into the tool.

Pre-Built Industry Standard Lifecycle Templates
Why reinvent the wheel? At your fingertips are the best-in-class industry recognized Lifecycle templates. Standard with QSM's SLIM-Estimate.

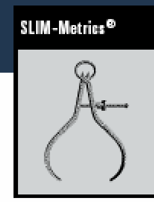
Industry Specific Knowledge Bases
Even when you have your own historic data, you can rely on SLIM-Estimate to empower you with credible historic data available only to QSM clients. It's the proven data you need. Tap into this knowledge.

Ready And Enabled Interfacing To Microsoft And Web
SLIM applications are designed to work in your office environment. When its time to create presentations or collaborate on the web, SLIM-Estimate lets you easily export data to your Microsoft Office or HTML formats.

Integrated Management Reporting
When reports and graphs aren't enough to show the entire picture, SLIM-Estimate's Smart Notes feature allows you to store relevant textual descriptions for use and reuse in an easy to find and intuitively linked manner. Next time you need a report, it will be at your fingertips.

Dynamic Solution Profiling
"What if?" is a question that often comes up. With SLIM-Estimate's Dynamic Solution Profiling, decision-makers can evaluate different options at the click of the mouse. Solutions change dynamically on the screen, allowing you to provide impressive alternatives quickly.

SLIM-Metrics®



Apply Projects' Past Performance Benchmarks to Achieve Future Measurable Success.

Question: Do you use historical data to improve the success ratio of your software development projects? You've probably heard that those who don't know the past are doomed to repeat it. Nothing rings more true when it comes to lack of applying sensible benchmarks that could have helped you avoid costly mistakes and achieve higher efficiencies. Until now, it was a difficult task. Seasoned project managers, just like you, have dreamed about having an intuitive, flexible, and expandable tool that enables you to easily perform historical data collection and analysis. Now it's here.

Introducing SLIM-Metrics from QSM…the only way to collect meaningful data for unlimited customized analyses. SLIM-Metrics is the proven tool of choice by organizations that want to stay ahead of the competition. From leading Fortune 500 companies to emerging-growth software companies, across government and manufacturing organizations, project managers harness the power of SLIM-Metrics to empower process improvements and enable improved productivity by applying the lessons of past project to optimize current and future project performance.

Get Your Data In Line With SLIM-Metrics.

Historical Data Capture
Store all of your organization's historical data in a single, open, relational database.

Customizable Database
Quickly extend your collected data to include any measures or metrics you like, and then use powerful graphing and statistical tools to analyze your data.

Fast, Efficient Data Entry
Set default values to describe your environment. The tabbed dialog interface makes data entry fast, consistent, and reduces time spent on error checking.

ODBC Compliant File Format
OPEN database allows you to create seamless interfaces to other ODBC-compliant tools.

Powerful Query Capability
Create specific subsets of your data and compare them to one another.

Supports Multiple Views
Analyze any measure or metric against any other(s) with unlimited charting/reporting views within a single workbook and customizable graph and report functions.

Industry Benchmarks
Industry trend lines for standard measures and metrics allow you to benchmark yourself against over 6,300 projects, or you can create your own trend lines from your completed project data.

Statistical Analysis Tools
Uncover relationships in your data with statistical and regression analysis features, including four different curve-fitting algorithms.

Compare up to 5 Datasets per Chart
Compare and contrast data from up to five different data sets on a single chart for internal benchmarking and competitive positioning.

Special Project Identification
Track a single project through multiple graphs to visualize trade-offs and cause/effect relationships.

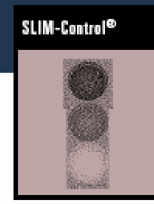
Point & Click Project Details
Identify or view details for any data point on a graph with a single mouse click.

Timesaving Features
Add custom notes, zoom a graph, read x-y coordinates, and configure charts/reports with just a few mouse clicks.

Presentation-Quality Outputs
SLIM-Metrics produces superb graphical outputs. You can easily print a set of graphs and tables that clearly makes your point. Bottom line: it's easier than ever to share your results.

Windows/MS Office Integration
Export features allow you to easily move your data, graphs, and tables into and out of other Windows or MS Office products. Bottom line: you don't have to change the way you work.

SLIM-Control®



It's Intelligent Reporting at Your Command.

Question: who truly controls the life cycle of your software development project? If you're like most seasoned managers, chances are the answer is NOT YOU. Unless you have a strategic software application that allows you to make sense out of your data, chances are that mission-critical criteria such as budgets, timelines, resource utilization, and forecasting are left to CHANCE. So software success is possible but not probable. That's until now.

If your data is overpowering you, now you can make a real paradigm shift with performance-driven software that PUTS YOU IN CONTROL of key decisions. Introducing SLIM-CONTROL from QSM. It's the intelligent way to harness critical data into intelligent reporting formats you can use to win. So software success is predictable and doable in real time. Any time. All the time.

SLIM-Control is the proven assessment tool of choice by organizations that can't afford to lose. From leading Fortune 500 companies to emerging-growth software companies, across government and manufacturing organizations, project managers harness the power of SLIM-Control to ensure projects meet budget and scheduling expectations. At inception. At mid way with mid-course corrections. In the predictable future.

Conquer The Data with SLIM-Control.

Earned Value Measurement Reports
Easily and seamlessly integrate critical reports to save time and money, and satisfy customers across multiple sectors to comply with regulatory and process requirements.

Adaptive Forecasting
Meet and exceed customers' expectations with reliable budgets and schedules created with the only software that learns your project's behavior and helps you adjust accordingly.

Management Scoreboard Reporting
View the status of your project at any time, create automated reports for top management, and simplify your post mortem analysis.

Embedded Variable Reporting
Save time and money with project reports that update automatically when new data is input.

Custom-Defined Metrics
Choose the metrics you want to capture and analyze, and create custom screens and dashboards that display only the information you need.

Control-Bound Rule Editor
Implement statistical process control and choose thresholds where management alarms should be sounded.

Plan & Forecast Logs
Quickly store and retrieve completed plans and forecasts for comparison, create an audit trail of the planning baseline, and simplify the post mortem process.

Industry Trend Lines
Rely on the proven performance trends of thousands of software developers to benchmark the success of your projects.

Application Programming Interface (API)
Ensure accurate results analysis and save time with automated data collection.

Import-Export
Analyze, update and share project results with easier importing and exporting of data with all MS Office products and MS Project.

Common User Interface
Simplify your process with a user interface that's common with all other products in the SLIM-Suite.

QSM®

r2 Estimator™

The r2Estimator is a Microsoft Windows®-based software decision-support tool offered by r2Estimating, LLC that implements the Ross Software Estimating Framework (rSEF). It helps users

- Estimate ***how much a project will cost***, ***how long it will take***, ***how many people will be required and when***, and ***how many defects will be delivered***.
- Estimate a project according to the relationships used in ***other software estimating models*** (facilitates ***crosschecking*** and ***validating*** existing estimates).
- Hierarchically ***structure projects*** according to the scope of each project element and families of elements.
- ***Interactively and dynamically*** examine all the possible outcomes of a project in terms of the ***confidence (probability of success)*** associated with each estimated value.
- Share and justify findings with a rich set of ***charts and reports***.

Feature Summary

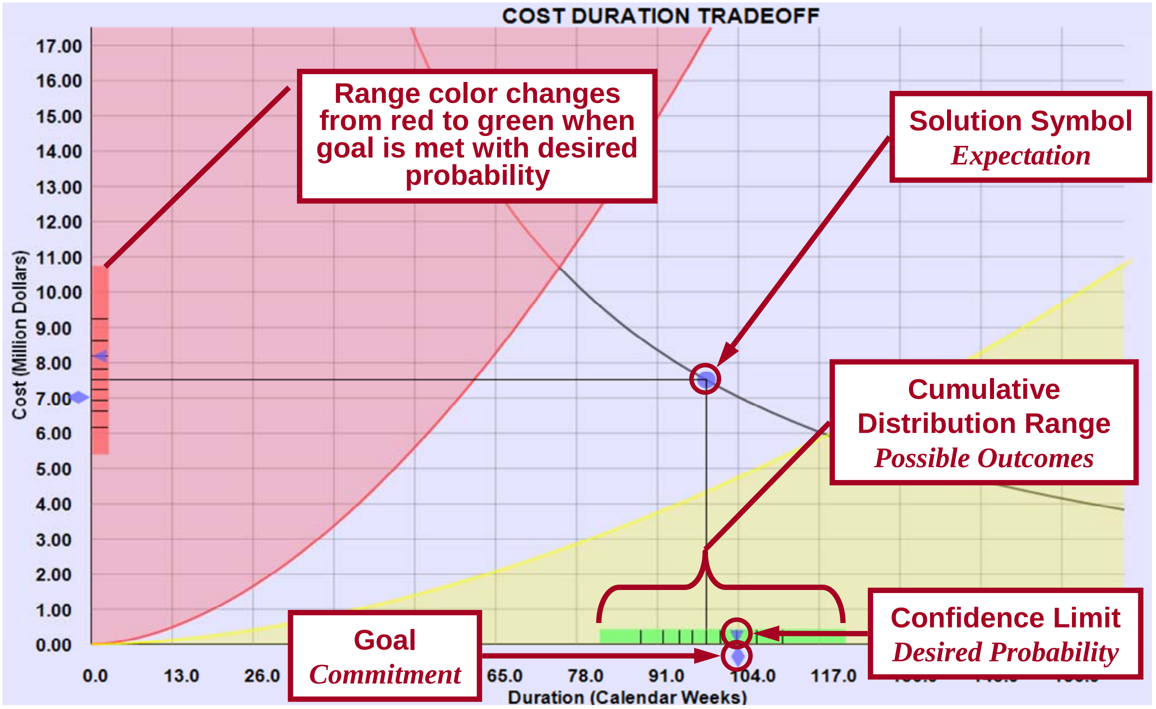
The r2Estimator

- Implements the rSEF set of equations that determine duration, effort, cost, staffing, and defects as a function of size, efficiency, management stress, and defect vulnerability.
- Manages a user-extensible set of development project ***categories***, each of which "coarse-tunes" the model for the type of software development being proposed by describing a specific instantiation of the rSEF equations based on either:
  - Regression analysis applied to some historical data set (e.g., r2 Database Avionics Software – ESLOC, ISBSG Database 4GL on Mainframe Platform – IFPUG UFP, etc.); or
  - The mathematical behavior of some commercially-available model for which the parameters and equations are in the public domain (e.g., COCOMO 81, COCOMO II, Jensen, Norden-Putnam-Rayleigh, etc.).
- Manages a user-extensible set of development ***profiles*** within each category, each of which "fine-tunes" the model within that particular category (e.g., Flight Controls as a profile within the r2 Database Avionics Software – ESLOC category).
- Includes COCOMO II parameter GUI for determining rSEF equation parameters as part of COCOMO II emulation.
- Includes Jensen Model parameter GUI for determining rSEF equation parameters as part of Jensen Model emulation.

- Allows the user to specify uncertainty associated with the independent variables (size, efficiency, and defect vulnerability) by triangular distribution [Lowest, Most Likely, Highest].
- Determines probability distributions associated with the dependent variables (duration, effort, cost, and defects) by Monte Carlo simulation at all levels of the project.
- Supports a hierarchical project structure (WBS) with drag and drop GUI that includes the following element types:
  - Project Summary Element
  - Summary Element
  - Decomposition Element
  - Construction Element
  - Integration Element
  - User Defined Task Element
  - Cost Only Element
  - Event (milestone) Element
- Allows the user to specify, for each element (except Cost Only and Event elements), the staffing profile shape as being either piecewise linear or Rayleigh with shaping control over each function including non-zero start and finish staff levels.
- Includes multiple notes logging (sequence, author, date, message text) for each element in the hierarchy.
- Displays three synchronized interactive Ross charts (probabilistic bivariate tradeoff for each of effort versus duration, cost versus duration, and defects versus duration) for all Construction Elements. Each Ross chart includes:
  - Tradeoff curve with limit regions
  - Drag-able solution symbol
  - Dynamic confidence range bars
  - Drag-able desired probability symbols
  - Drag-able goal symbols
  - Hover metrics display on all drag-able items
  - Right-click pop-up data entry dialog boxes for all drag-able items
  - Scalable axes

- Displays an interactive Gantt chart for each non-Construction element showing its parent and all of its parent's offspring. Each Gantt chart includes:
  - Drag-able element schedule bars
  - Confidence range bars
  - Drag-able desired probability indices
  - Drag-able goal indices
  - Hover metrics display on all drag-able items
  - Right-click pop-up data entry dialog boxes for all drag-able items
  - Scalable axes
- Displays a staffing chart for all non-construction elements, date-synchronized with its associated Gantt chart. Each staffing chart includes:
  - Required staffing curve
  - Available staffing curve
  - Scalable axes
- Displays tab-organized metrics reports and charts including:
  - Project results
  - Element metrics
  - Element inputs
  - Notes
  - Duration CDF (confidence probability versus duration value)
  - Effort CDF (confidence probability versus effort value)
  - Cost CDF (confidence probability versus cost value)
  - Delivered defects CDF (confidence probability versus delivered defects value)
  - Staffing
- Produces graphics that are copy-able to other Microsoft Windows applications.
- Produces user-definable XML-formatted metrics reports.
- Produces and maintains XML-formatted Project Files and Default Files.
- Provides export of a Project File’s WBS to Microsoft Project®.

Ross Charts



Example Ross Chart (Cost versus Duration)

r2Estimator implements a new chart called a ***Ross Chart*** that is a graphical display of the confidence (probability of success) and goal satisfaction of two correlated random variables. Ross Charts consist of:

- A two-dimensional Cartesian axis and coordinate system;
- A line or curve representing the correlation (relationship) between the two random variables;
- Indication(s) of the relationship's limit(s) (reasonable range);
- Interactive dynamic solution symbol on the relationship curve representing a specific instance (solution) of the relationship;
- Dynamic projection lines from the solution symbol to each axis;
- Dynamic cumulative distribution range symbol on the axis-ends of each projection line, each range symbol indexed in increments of 10% confidence probability and representing its corresponding random variable's cumulative distribution function (CDF);
- Interactive dynamic confidence limit (risk tolerance) symbol on each cumulative distribution range symbol;
- Interactive dynamic goal symbol on each axis representing the goal/commitment value associated with the corresponding variable.

Ross Software Estimating Framework

The Ross Software Estimating Framework (rSEF) is a set of general software effort, duration, and defects estimating relationships that are based on the notion that software construction is the application of effort (labor) over some duration (period of elapsed calendar time) that produces a desired software product (size) and undesired byproducts (defects). The fundamental rSEF relationships are

Software Productivity Law

$$Effort^{(\alpha_E)} \times Duration^{(\alpha_t)} = \frac{Size}{Efficiency}$$

Defect Propensity Law

$$\frac{Effort^{(\varphi_E)}}{Duration^{(-\varphi_t)}} = \frac{Defects}{Defect\ Vulnerability}$$

Management Stress Law

$$Management\ Stress = \frac{Effort}{Duration^{(\gamma)}}$$

Brooks' Law (Limit) – Minimum Time

$$Management\ Stress_{\max} \geq \frac{Effort}{Duration^{(\gamma)}}$$

$$\therefore Management\ Stress_{\max} = \frac{Effort_{t\min}}{Duration_{\min}^{(\gamma)}}$$

Parkinson's Law (Limit) – Minimum Effort

$$Management\ Stress_{\min} \leq \frac{Effort}{Duration^{(\gamma)}}$$

$$\therefore Management\ Stress_{\min} = \frac{Effort_{\min}}{Duration_{E\min}^{(\gamma)}}$$

rSEF Parameters Included in Category Specification

- Effort Exponent α_E
- Duration Exponent α_t
- Defect Effort Exponent φ_E
- Defect Duration Exponent φ_t
- Gamma γ
- Minimum Management Stress $M_{\min}$
- Nominal (Typical) Management Stress M_{nom}
- Maximum Management Stress $M_{\max}$

- Efficiency Scale Vector $\hat{\boldsymbol{\eta}}$ (Efficiency values from -3 standard deviations to +3 standard deviations in increments of 0.5 standard deviations)
- Defect Vulnerability Scale Vector $\hat{\boldsymbol{\delta}}$ (Defect Vulnerability values from -3 standard deviations to +3 standard deviations in increments of 0.5 standard deviations)

rSEF Parameters Included in Profile Specification

- Efficiency 3-point Estimate $\boldsymbol{\eta} = \left[\eta_{Lowest}, \eta_{Most\ Likely}, \eta_{Highest}\right]$

Defect Vulnerability 3-point Estimate $\boldsymbol{\delta} = \left[\delta_{Lowest}, \delta_{Most\ Likely}, \delta_{Highest}\right]$

PRICE® True S™ Model

Model Overview

PRICE® True S™, introduced by PRICE Systems in 2003, is a parametric-based predictive model. Built on the same core methodology of its predecessor, PRICE S, PRICE® True S applies a comprehensive approach to estimating software size, reuse, productivity, and the true cost of integrating COTS. PRICE® True S™ calculates the effort of each resource consumed by the software development, acquisition, and support activities and produces an estimate that aligns activity and resource costs with real world practices.

PRICE© True S calculates the predicted cost of software development and support projects as effort (hours, weeks, months, or currency) and estimates a typical schedule. PRICE® True S ban be used to credibly predict cost and schedule when information is limited (such as during the system concept phase of a project).

PRICE® True S is applicable to all types of software projects, including project planning, proposal preparation, proposal evaluation, bid and no-bid decisions. The model distributes estimated costs and labor requirements over time to enable budget planning, and provides model calibration and uncertainty analysis tools.

Four different sizing tools enable cost estimators to estimate software size based on source lines of code (SLOC), function points (FPs), use case conversion points (UCCPs), or predictive object points (POPs). When estimating COTS components as part of a software project, the Functional Size sizing tool is available to estimate the magnitude of the component based upon the requirements it must meet. Systems that include COTS components also require a Glue Code Size value to define the amount of code that must be written to incorporate the COTS component into the system (such as, to provide interfaces, interpret returns codes, translate data to proper formats, and to compensate for inadequacies or errors in the selected COTS component).

The PRICE® True S™ model provides advanced project planning, estimating, management, and control capabilities that:

- Estimate system-level costs, resources, and schedule for integrated hardware, software, and IT projects
- Account for project level costs of managing and integrating multiple system components
- Estimate software size in categories of new, reused, adapted, deleted, auto-generated, and auto-translated
- Credibly account for the true cost of selecting, acquiring, updating, and integrating COTS (Commercial Off-The-Shelf) software and COTS-intensive systems
- Quantify the characteristics that affect organizational productivity, such as CMM level, the use of IPTs, and collaboration
- Analyze resource capacity and utilization
- Create, preserve, apply, and document multiple sets of custom inputs and calibration settings to explore *what-if* scenarios
- Perform management tasks such as authorizing users, maintaining projects, allowing server connections, and capturing organization workflow
- Import and export from PRICE S and third party software like Microsoft Project and ACEIT

System Level Estimating

PRICE© True S™ is one of several activity-based predictive models that represent the most recent generation of commercial models developed by PRICE Systems, L.L.C. The PRICE® TruePlanning® proprietary framework supports interoperability of multiple, industry-specific models to create system-level estimates. TruePlanning employs universal assemblies to account for both technical and organizational overhead of diverse projects by rolling up costs and schedules simultaneously from software, hardware, and IT cost models. The client server version of PRICE® TruePlanning® incorporates project management tools, project sharing, workflow management, and other multi-user capabilities.

PRICE® True S™ Cost Methodology

PRICE® True S™ is a collection of predictive models (cost objects) that simulate the activities and resources required for software project development and support. Assembly-level cost objects model the activities that comprise the technical and organizational project level tasks of software projects, such as: requirements definition and analysis, system design, integration and test, verification and validation, planning and oversight, management and control, quality assurance, configuration management, vendor management, and documentation.

Each PRICE cost model is defined by a collection of dynamic cost estimating relationships. These PRICE relationships are not bounded by a single database; nor does the PRICE cost model depend on a single set of fixed CERs.

The PRICE cost methodology is best understood as a reality ring like the one shown in Figure 1. The center of the ring represents the core cost estimating relationships of the cost model. The core CER is based on software size and productivity. It represents software development in an ideal world. Outer rings represent additional cost-to-cost relationships, CERs, and other cost factors that adjust the core cost to account for the realities and complications of a real world project. PRICE® True S™ actually contains thousands of mathematical equations that relate input parameters to cost.

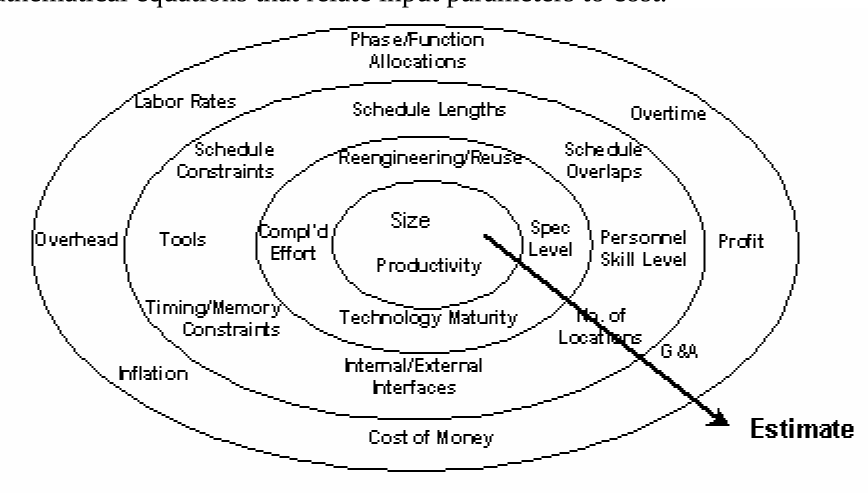


Figure 1: PRICE Cost Methodology

The core equation calculates the amount of effort based on the consumption of resources by each activity. A description of the form of this fundamental relationship follows.

EFFORT = *Size*Baseline Productivity*Productivity Adjustments*

Where:

Size = Expressed in SLOC, FPs, POPs, or Use Case Conversion Points (UCCPs)

Baseline Productivity = an industry standard

Productivity Adjustment = the effects of cost drivers on productivity

PRICE® True S™ starts with an industry ideal productivity baseline and proceeds to adjust to an expected productivity of each activity based on cost driver inputs that describe the unique conditions of the software project in question. The effort calculation establishes the activity requirements that are used to calculate the optimal team size and resource requirements.

RESOURCE REQUIREMENTS = ***Activity Requirements* f(size, functional characteristics, operating specification)***

Where:

Resource Requirements = *the number of labor hours*

Activity Requirements = *the number of activity units that are required*

Size = Expressed in *SLOC, FPs, POPs, or UCCPs*

Functional Complexity = *quantification of code complexity*

Operating Specification = *operating environment specified by the customer*

SCHEDULE = ***Effort/Optimal Team Size***

Where:

Optimal Team Size = *f(size, functional complexity, operating specification)*

The calculated values for effort, team size, and schedule are determined by cost driver inputs that describe the software development project. PRICE® True S™ calculates a reference schedule based on ideal conditions. Inputs for *schedule multipliers* adjust the schedule to reflect performance history; inputs for the *schedule effect* parameters control the application of penalties for schedule compression and expansion.

PRICE® True S™ Principal Inputs

A listing of the principal PRICE® True S™ cost drivers that adjust software development productivity follows.

- ***Software Size***.
- ***Amount of Reuse***
- ***Functional Complexity***
- ***Operating Specification***
- ***Project Constraints***
- ***Programming Language/ Implementation Tool***
- ***Development Team Complexity***
- ***Security Process Level and Security Level***
- ***Organizational Productivity***
- ***Project Complexities***
- ***Multiple Site Development***
- ***Development Process***
- ***Internal Integration Complexity***
- ***COTS Component Evaluation Process***
- ***COTS Integration Team Maturity***
- ***Vendor and Product Complexity***
- ***Upgrade Frequency***
- ***Tailoring Complexity***
- ***Glue Code Programming Language***

PRICE® True S™ Support Cost Methodology

PRICE® True S™ calculates maintenance costs and labor requirements from acquisition and deployment data for software and COTS components. Assembly-level deployment and retirement dates establish the boundaries of the maintenance period for software and COTS components. PRICE® True

S™ profiles maintenance activity costs and labor requirements over time based upon the number of anticipated software installations and the selected distribution. The model also accounts for the impact that software maintenance activity has on planning and oversight activities, including project management, quality assurance, configuration management, and documentation.

PRICE® True S™ Calibration

The cost estimator controls much of the cost estimating process through calibration and by overriding the preset values of the PRICE cost models. Calibration enables cost estimators to accurately characterize an organization's cost allocation practices and to improve cost estimates by incorporating empirical complexity values that are specific to a software project. The PRICE reality ring process described in the previous sections also describes the PRICE calibration methodology. When known values (recorded cost) are available for the purpose of calibration, the model can better deduce the complexities that affect productivity and are inherent to an organization's software development experiences. A second level of calibration enables cost estimators to match the software model cost allocations to actual experience.

PRICE® True S™ Outputs

PRICE® True S™ offers multidimensional views of project data, including inputs and calculated outputs. The tracking and reporting features of PRICE® True S™ enable cost estimators to document the source and rationale for these custom inputs and to preserve a record of all inputs to the project, including throughput costs that were not calculated by the PRICE model. PRICE® True S™ collects input information in the Project Audit report and can populate the U.S. government's Exhibit 300 report with text and project data.

The model provides total system-level cost, incorporating preset or user-defined rates for labor, indirect costs, and escalation. Customizable project settings enable cost estimators to view results as *constant year* or *as spent* costs.

Calculated labor requirements and resource capacity profiles can be viewed in hours, week, months, or currency units. Cost estimators can tailor schedule effects, constraining or eliminating schedule penalties, and adjust or eliminate activities and resources.

In addition to labor and cost profiles, PRICE® True S™ calculates maintenance cost per size unit, total maintenance hours, and total maintenance cost. Both graphical charts and text-based upon output reports are available. Project results can be copied and pasted to third-party programs, and exported to a variety of file formats, including formats compatible with Microsoft Project and ACEIT. Cost Risk Analysis is also incorporated within PRICE® True S™.

PRICE® H™ and PRICE® True H™ Models

Overview of Features

PRICE® H was the first commercially-available hardware acquisition and development cost model in 1975. For more than 30 years its capabilities have been continuously updated by ongoing cost research and the experience gained from thousands of federal and commercial projects. PRICE® True H™, developed in 2005, is one of several predictive models that represent the most recent generation of commercial models from PRICE Systems, L.L.C. The PRICE® TruePlanning® proprietary framework supports interoperability of these industry-specific models, rolling up cost and schedule from software, hardware, and IT cost models to generate a system-level project estimate. Interoperability with other models is the key to planning and managing system-of-systems (SoS) projects. The client server version also incorporates project management features, project sharing, and other multi-user capabilities. Estimates produced by PRICE® H™ and PRICE® True H™ are based on the same core cost methodology.

PRICE® H™ and PRICE® True H™ estimate the cost, resources, and schedule required to develop, produce, modify, integrate, and test hardware systems, assemblies, subassemblies, and single components (including purchased and furnished items).

Both models offer the flexibility to produce rough order of magnitude (ROM) and detailed budget estimates for development and production, development only, paper studies (no hardware built), production only, and development in production (no prototypes built). The models spread estimated costs and labor hours or months over time to enable budget planning, and both support comprehensive model calibration and rigorous risk analysis.

PRICE® H™ and PRICE® True H™ include methods to estimate single hardware items as well as complex systems of hardware items and/or hardware subsystems and systems. Among the capabilities are: integration and management of subcontracted and COTS hardware; identification of production costs by lot; schedule costs assessment; systems integration and project level cost identification; integration with third party spreadsheet, project management, and engineering software; client/server version installation with associated client management capability.

Functional Relationships and Methodology

The PRICE® H™ and PRICE® True H™ models estimate hardware cost from quantitative parameters (such as, manufacturing quantities, weight, and size); qualitative parameters (such as, specification level, equipment function, and level of integration), and schedule-driving parameters (such as, the number of months to the first prototype, the manufacturing rate, and the amount of new design). PRICE models can deduce input values when a limited amount of hardware information is available (during the concept phase, for example).

Each PRICE cost model is defined by a collection of dynamic cost estimating relationships, including non-cost to cost relationships and cost-to-cost relationships. The PRICE relationships are not bounded by a single database and the PRICE cost models do not depend on a single set of fixed CERs. The most effective implementation of the PRICE model relies on the database of the user—through calibration.

The PRICE cost methodology is best understood as a reality ring like the one shown in Figure 1. The center of the ring represents the core cost estimating relationships of the cost model. Outer rings represent additional cost-to-cost relationships, CERs, and other cost factors that adjust the core cost to account for the realities and complications inherent in a real project. The diagram in Figure 1 provides a simplified view; both PRICE hardware cost models actually contain thousands of mathematical equations that relate input parameters to cost.

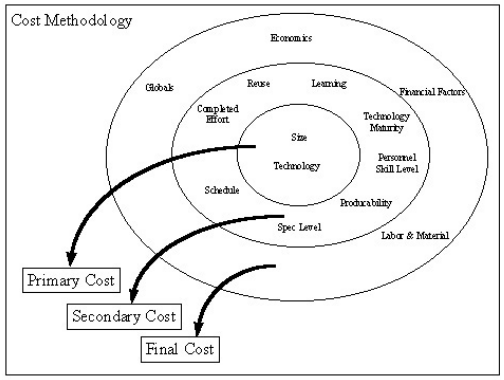


Figure 1: PRICE Cost Methodology

The PRICE hardware models generate a cost estimate in three steps:

Step 1: The primary cost estimate is developed from core CERs based on weight and manufacturing complexity. Manufacturing complexity measures the technology, producibility, yield, labor, and materials that are required to produce a hardware item. This complexity index provides a *measure* of the cost per weight unit (*cost density)*.

Step 2: Moving from the ideal world at the core to the reality of the world, step 2 applies cost-to-cost relationships and additional CERs to adjust the core cost estimate to account for design reuse, specification level, technological maturity, the amount of new design effort, the degree of automation in production processes, and other relevant factors that describe actual project complications.

Step 3: The final stage splits the estimate into labor, material, and other direct costs and applies economics, labor rates, and the organizational burdening to generate an estimate that matches the cost estimator's experience.

The information shown in Table 1 summarizes the key drivers and fundamental input parameters.

Weight and Manufacturing Complexity: The most crucial PRICE input parameters		
	Electronics	**Non-Electronics**
Weight	Module weight of components, connectors and board.	Chassis, antennae, optics, motors, engines, precision assemblies, etc.
Manufacturing Complexity	Measure of cost per weight unit based on technology and function.	Measure of cost per weight unit based on material, function, precision and others.

Fundamental input parameters to PRICE hardware models
Quantities of equipment developed, produced, modified, purchased, integrated, and tested; Amount of design inventory and complexity of development engineering; Operating environment and hardware specification requirements; Schedules for development, production, procurement, modification, integration, and testing; Production fabrication process; Pertinent labor, overhead, material burden, and escalation rates; cost mark-ups for G&A fees; profit; and cost of money; Factors that profile the organization's financial and labor accounting.

Table 1 Principle PRICE Hardware Model Inputs

Calibration Methodology

Calibration enables cost estimators to accurately characterize an organization's cost allocation practices and to improve cost estimates by incorporating empirical complexity values that are specific to the hardware.

The PRICE reality ring process defines the PRICE calibration methodology. When product experience (recorded cost) for a completed item is available, a product complexity value can be deduced by using only the characteristics that are responsible for the difficulty inherent to designing or building the item. Cost density is computed after the reality ring relationships that are not product-specific are measured.

A second level of calibration enables cost estimators to match the hardware model cost allocations to actual experience. This is achieved by producing a default allocation with calibrated complexities and then comparing the defaults to previously recorded data. Next, the default results are re-allocated by using simple scalar factors to match the historical data. Both the calibrated complexities and allocation scalars become data points within a PRICE knowledge base. These knowledge bases customize PRICE and support the credibility of the estimates that rely on them.

Both PRICE® H™ and PRICE® True H™ provide the capability to calibrate the model based on historical cost and past performance. PRICE® H™ provides a special Calibration item for this purpose. Data from the Calibration item can be copied to an appropriate element item in the WBS. PRICE® True H™ provides a Calibration tool that preserves multiple calibration records for editing and reuse.

Risk Analysis Methodologies

The two PRICE hardware models employ different risk methodologies for cost risk analysis.

PRICE® H™ randomly samples the selected statistical distribution (normal, triangular, beta, or uniform), based on the uncertainty specified by the cost estimator, and calculates a separate cost estimate for each iterative pass. This Monte Carlo approach generates a cumulative probability distribution graph that displays the probability of cost and schedule overrun (and underrun) as well as a tabular risk analysis report that displays cost and schedule estimates in 5% increments of confidence, statistical figures of merit, and the specified uncertainty for each input parameter.

PRICE® True H™ cost risk analysis is based on FRISK, a methodology widely used in the U.S. aerospace industry. This methodology recognizes the degree of correlation between cost items as a significant driver of uncertainty. PRICE® True H™ offers the ability to set correlation values to describe this interdependence. The True H model calculates uncertainty based on project settings for project phase and technology maturity and applies a triangular distribution based on likely, optimistic, and pessimistic values for each risk driver (preset, auto-calculated, or user-selected).

Output Information

Both PRICE hardware models provide multidimensional views of project data, including inputs and calculated outputs.

Cost estimators can document their work with reports that describe all project inputs, including user-defined values, derived inputs, default tables and multipliers, and the cost estimator's custom notes documenting the rationale for input decisions. PRICE® True H™ collects input information in the Project Audit report, and can populate an Exhibit 300 report with text and project data.

Cost estimators can customize report formats to include program cost and schedule for development and production, development only, production only, summary totals, cost allocations, labor hours and material costs, multiple lot totals, expenditure and labor profiles (spread over time) for multiple reporting periods, and risk analysis data. By using analyst-controlled rates for labor and indirect costs, the model also provides total cost for the elements contained in the WBS. Project settings enable the analyst to view *constant year* or *as spent* project costs. Both hardware models convert all cost estimates to a single output currency. Cost estimators can tailor schedule outputs as well. For example, estimators can opt to take into account interdependent elements when calculating Integration and Test schedule dates and they can constrain or eliminate schedule penalties. Graphical charts and text based output reports are available.

APPENDIX B

Detailed Math of Cost Estimating Relationships

The Detailed Math of Cost-Estimating Relationships

I. INTRODUCTION AND OVERVIEW OF CER DEVELOPMENT PROCESS

The Cost Estimating Relationship (CER) is the distinguishing feature of parametric estimating. A CER is a mathematical expression, which describes how the values of, or changes in, a "dependent" variable are partially determined, or "driven," by the values of, or changes in, one or more "independent" variables. The CER defines the relationship between the dependent and independent variables, and describes how it behaves. Since a parametric estimating method relies on the value of one or more input variables, or parameters, to estimate the value of another variable, a CER is actually the quintessential parametric estimating technique.

The dependent variable of a CER is cost or a closely related resource measure such as staff hours. The independent variable or variables are typically technical parameters that have been demonstrated to correlate with cost. In a cost-to-cost relationship, the independent variables are also costs – examples are CERs which use manufacturing cost to estimate quality assurance cost, or to estimate the cost of expendable material such as rivets, primer, or sealant. The cost of one element is used to estimate, or predict, that of another. In a non cost-to-cost relationship, the CER uses a characteristic of an item to predict its cost. Examples are CERs that estimate an item's manufacturing costs based on its weight (independent variable), or the design engineering costs from the number of engineering drawings (independent variable) involved.

It is important to note that the term "cost driver" is meant in a fairly broad sense, to include cases like those above where the "independent" variable does not actually cause the "dependent" variable to be what it is. But the two variables may be sufficiently correlated with (or "track") each other such that if one is known or estimated, then the other can be known or estimated fairly well. Thus, in the cost-to-cost relationship example above, the size, quantity and complexity of the item being produced may be the real cost drivers of both the manufacturing costs <u>and</u> the quality assurance costs. The design engineering CER example illustrates true cause-and-effect behavior, where the design-engineering costs are <u>caused</u> to be what they are by the number of drawings required.

The manufacturing cost CER example is a little murkier. The item's weight and cost may correlate well, but the weight is not exactly the cause for the cost to be what it is. It is usually the basic requirements that the item must satisfy which drive both cost and weight (or size). In fact, if the requirements dictate that the item's weight be limited to the extent that unusually expensive production methods must be used, then weight per se and cost may have an inverse (i.e., negatively correlated) relationship.

Regardless of the underlying cause and effect relationships, in the context of this chapter CER cost drivers are assumed to be either true drivers of cost or surrogates for the true cost driving requirements and constraints on the item being estimated. In many cases weight may be viewed as a good representative for most of the requirements that drive cost. In other cases it may represent cost driving requirements poorly – particularly in cases where smallness or lightness are at a premium. The same might be true for other variables that represent size or magnitude of the cost element being estimated, such as software source lines of code or processing throughput.

A CER is a valuable estimating tool and can be used at any time in the estimating process. For example, CERs may be used in the program concept or validation phase to estimate costs when there is insufficient system definition for more detailed approaches, such as the classical “grass roots” or “bottoms-up” methods. CERs can also be used in a later phase of a program as primary estimates or as crosschecks of non-parametric estimates. CERs may also form the primary Basis of Estimate for proposals submitted to the Government or higher-tier contractors. They are also used extensively by government agencies to develop Independent Cost Estimates for major elements of future programs. In practice, a reasonable estimate of a program can sometimes be made parametrically with as little as one simple CER consisting of a single independent variable. As the program definition is fleshed out, additional parameters become available for use in cost estimation. Parametric cost models comprised of several CERs can then give estimates at lower levels of definition. Before developing complex parametric models, analysts typically create simple CERs which demonstrate the utility and validity of the basic parametric modeling approach to company and Government representatives.

The proper development and application of CERs depends on the collection and preparation of data on historical programs and on applying appropriate mathematical and statistical techniques. Figure1 is a schematic of the CER development process. This chapter explains each step of this process. The next section describes data collection and analysis. Topics include types and sources of data, collection methods, and evaluation and normalization to ensure homogeneity. The third section presents the methods for developing CERs. In particular, the identification of cost drivers is described along with challenges in dealing with limited data sets. An overview of the most mathematically correct techniques for regression and curve fitting is included. CER selection is the subject of the fourth section which includes a table of simple CERs that are useful as rules of thumb. The fifth section covers calibration and validation of the CERs. The sixth and final section relates common issues and pitfalls in developing CERs.

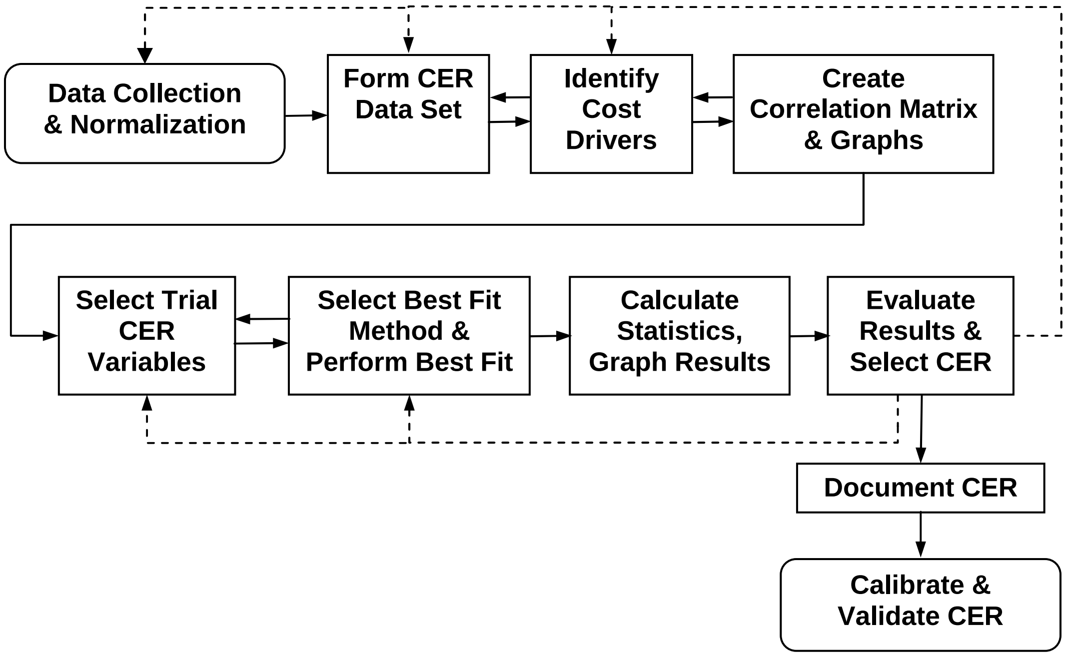


Figure 1: CER Development Process

II. DATA COLLECTION

All parametric estimating techniques, including CERs and complex models, require credible data before they can be used effectively. This section discusses the processes needed to collect and analyze the data used in parametric applications, as well as data types, sources, and adjustment techniques. It also:

- Identifies sources of information that can be collected to support data analysis activities.
- Describes various methods of adjusting raw data to put it on a common basis (i.e., data normalization).
- Discusses the importance of collecting historical cost and non-cost (e.g., technical or programmatic) data to support parametric estimating techniques.

A. DETAILED DISCUSSION

Data Types and Collection

Parametric techniques require the collection of historical cost data (including labor hours) and the associated non-cost information and factors that describe and strongly influence those costs. Data should be collected and maintained in a manner that provides a complete audit trail with expenditure dates so that costs can be adjusted for inflation. Non-recurring and recurring costs should be separately identified. While there are many formats for collecting data, one commonly used by industry is the Work Breakdown Structure (WBS), which provides for the uniform definition and collection of cost and certain technical information. DoD

Handbook – MIL-HDBK-881A provides detailed guidance on the use of WBS. Regardless of the method, a contractor's data collection practices should be consistent with the processes used in estimating, budgeting, and executing the projects from which the data was collected. If this is not the case, the data collection practices should contain procedures for mapping the cost data to the cost elements of the parametric estimating technique(s) which will be used.

The collection point for cost data is generally the company's management information system (MIS), which in most instances contains the general ledger and other accounting data. All cost data used in parametric techniques must be consistent with, and traceable to, the collection point. The data should also be consistent with the company's accounting procedures and generally accepted cost accounting practices.

Technical non-cost data describe the physical, performance, and engineering characteristics of a system, sub-system or individual item. For example, weight is a common non-cost variable used in CERs and parametric estimating models. Other examples of cost driver variables are horsepower, watts, thrust, and lines of code. A fundamental requirement for the inclusion of a technical non-cost variable in a CER is that it must be a significant predictor of cost. Technical non-cost data come from a variety of sources including the MIS (e.g., materials requirements planning (MRP) or enterprise resource planning (ERP) systems), engineering drawings, engineering specifications, certification documents, interviews with technical personnel, and through direct experience (e.g., weighing an item). Schedule, quantity, equivalent units, and similar information come from industrial engineering, operations departments, program files, or other program intelligence.

Other generally available programmatic information that should be collected relates to the tools and skills of the project team, the working environment, ease of communications, and compression of schedule. Project-to-project variability in these areas can have a significant effect on cost. For instance, working in a secure facility under "need to know" conditions or achieving high levels in various team certification processes can have a major impact on costs.

Once collected, cost data must be adjusted to account for the effect of certain non-cost factors, such as production rate, improvement curve, and inflation - this is data normalization. Relevant program data including development and production schedules, quantities produced, production rates, equivalent units, breaks in production, significant design changes, and anomalies such as strikes, explosions, and other natural disasters are also necessary to fully explain any significant fluctuations in the data. Such historical information can generally be obtained through interviews with knowledgeable program personnel or through examination of program records. Fluctuations may exhibit themselves in a profile of monthly cost accounting data; for example, labor hours may show an unusual "spike" or "depression" in the level of charges. Later in this chapter is a description of the data analysis and normalization processes.

Data Sources

The specification of an estimating methodology is an important step in the estimating process. The basic estimating methodologies (analogy, grassroots, standards, quotes, and parametric) are all data-driven. Credible and timely data inputs are required to use any of these methodologies. If data required for a specific approach are not available, then that estimating methodology cannot be used. Because of this, the estimator must identify the best sources for the method to be used.

Figure 2 shows nine basic sources of data and whether they are considered a primary or secondary source of information. When preparing a cost estimate, estimators should consider all credible data sources; whenever feasible, however, primary sources of data have the highest priority of use.

Primary data are obtained from the original source, and considered the best in quality and the most reliable. Secondary data are derived (possibly "sanitized") from primary data, and are not obtained directly from the source. Because of this, they may be of lower overall quality and usefulness. The collection of the data necessary to produce an estimate, and its evaluation for reasonableness, is critical and often time-consuming. Collected data includes cost, program, technical, and schedule information because these programmatic elements drive those costs. For example, assume the cost of an existing program is available and the engineers of a new program have been asked to relate the cost of the old to the new. If the engineers are not provided with the technical and schedule information that defines the old program, they cannot accurately compare them or answer questions a cost estimator may have about the new program's costs. The bottom line is that the cost analysts and estimators are not solely concerned with cost data - they need to have technical and schedule information to adjust, interpret, and support to the cost data being used for estimating purposes. The same is true of programmatic data when it affects costs. As an example, assume that an earlier program performed by a team at CMMI (Capability Maturity Model Integration) level 2 is to be compared to a new program where the team will be at CMMI level 4. The expectation is that the CMMI level 4 team will perform much more efficiently than the level 2 team.

Sources of Data	
Source	**Source Type**
Basic Accounting Records	Primary
Cost Reports	Either (Primary or Secondary)
Historical Databases	Either
Functional Specialist	Either
Technical Databases	Either
Other Information Systems	Either
Contracts	Secondary
Cost Proposals	Secondary

Figure 2: Sources of Data

A cost estimator has to know the standard sources of historical cost data. This knowledge comes both from experience and from those people capable of answering key questions. A cost analyst or estimator should constantly search out new sources of data. A new source might keep cost and technical data on some item of importance to the current estimate. Internal contractor information may also include analyses such as private corporate inflation studies, or "market basket" analyses (a market basket examines the price changes in a specified group of products). Such information provides data specific to a company's product line, but which could also be relevant to a general segment of the economy. Such specific analyses would normally be prepared as part of an exercise to benchmark government provided indices, such as the consumer price index, and to compare corporate performance to broader standards.

Some sources of data may be external. This includes databases containing pooled and normalized information from a variety of sources (e.g., other companies, public record information). Although such information can be useful, it may have weaknesses:

- No knowledge of the manufacturing and/or software processes used and how they compare to the current scenario being estimated.
- No knowledge of the procedures (e.g., accounting) used by the other contributors.
- No knowledge on the treatment of anomalies (how they were handled) in the original data.
- The inability to accurately forecast future indices.

Sources of data are almost unlimited, and all relevant information should be considered during data analysis. Figure 3 summarizes the key points about data collection, evaluation, and normalization.

Data Collection, Evaluation and Normalization
Very Critical Step
Can Be Time-Consuming
Need Actual Historical Cost, Schedule, and Technical Information
Know Standard Sources
Search Out New Sources
Capture Historical Data
Provide Sufficient Resources

Figure 3: Data Collection, Evaluation & Normalization

Routine Data Normalization Adjustments

Cost data must be adjusted to eliminate any bias or “unevenness” which other factors may cause in it. This is called normalization and is intended to make the data set homogeneous, or consistent. The analyst needs to examine every data set to ensure it is free from the effects of:

- the changing value of the dollar over time
- cost improvement as the organization improves its efficiency

- various production quantities and rates during the period from which the data were collected

Non-recurring and recurring costs are also segregated as part of the normalization process.

Figure 4 shows the typical data normalization process flow. This does not describe all situations, but does depict the primary activities followed in data normalization.

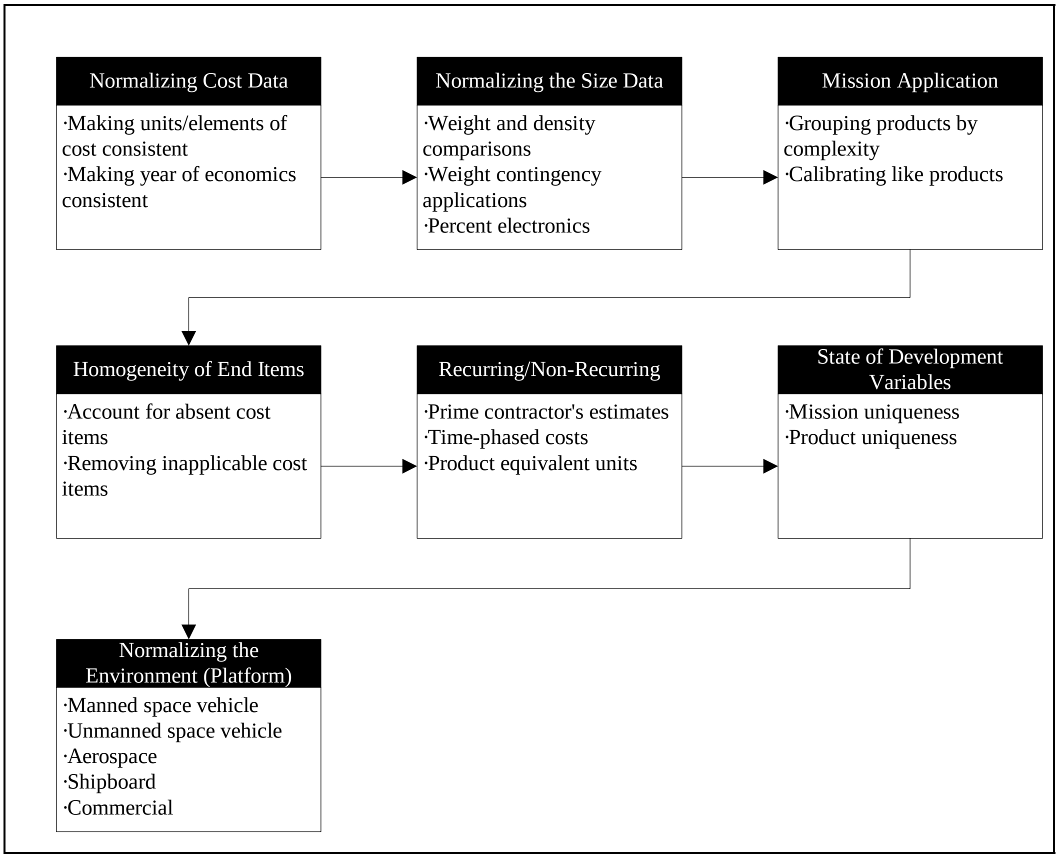


Figure 4: Data Normalization Process Flow

Some data adjustments are routine in nature and relate to items such as inflation. These are discussed below. Other adjustments are more complex in nature (e.g., relating to anomalies), and will be discussed later.

Inflation

Inflation is defined as a rise in the general level of prices, without a rise in output or productivity. There are no fixed ways to establish universal inflation indices (past, present, or future) that fit all possible situations. Inflation indices generally include internal and external information and factors (such as Section II discusses). Examples of external information are: the Consumer Price Index (CPI), Producer Price Index (PPI), and other forecasts of inflation from various econometric models. Therefore, while generalized inflation indices may be used, it may also be possible to tailor and negotiate indices used on an individual basis to specific labor rate agreements (e.g., forward pricing rates) and the actual materials used on a project. Inflation indices should be based on the cost of materials and labor on a unit basis (e.g., pieces, pounds, hours), and should not include other considerations such as changes in manpower loading, or the amount of materials used per unit of production. The key to inflation adjustments is consistency. If cost is adjusted to a fixed reference date for calibration purposes, the same type of inflation index must be used in escalating the cost forward or backwards from the reference date, or to the date of the estimate.

Non-Recurring and Recurring Costs

The prediction of system acquisition costs requires that non-recurring and recurring costs be separately estimated.

Non-recurring costs include all the efforts required to develop and qualify a given item, such as requirements definition/allocation, design, analysis, development, and qualification/verification. Manufacturing and test of development (breadboard and engineering) units, qualification units, and life test units are typically included in the non-recurring cost of hardware end items. Retrofitting and refurbishment of development hardware for requalification is also treated as non-recurring. Virtually all software development and testing costs prior to initiation of routine system operation are non-recurring. Non-recurring integration and test efforts usually end when qualification tests are complete. The non-recurring portions of services costs and some hardware end item costs, such as engineering, are commonly defined as those which take place prior to and during Critical Design Review (CDR). Development, acquisition, production, and checkout of all tooling, ground handling, and support equipment, test equipment, test software, and test procedures are also usually classified as non-recurring.

Recurring costs cover all the efforts required to produce end-item hardware, including manufacturing and test, engineering support for production, and spare units or parts. Recurring integration and test efforts include integration of production units and acceptance testing of the resulting assemblies at all levels. Refurbishment of hardware for use as operational or spare units is usually recurring. Maintenance of test equipment and production support software costs are commonly classified as recurring, while maintenance of system operational software, although recurring in nature, is often considered part of Operating and Support costs (which may also have nonrecurring components).

Cost Improvement Curve

When first developed, cost improvement was referred to as "learning curve" theory, which states that as the quantity of a production item doubles, the manufacturing hours per unit expended producing it decrease by a constant percentage. The learning curve, as originally conceived, analyzed labor hours over successive production units of a manufactured item, but the theory behind it has now been adapted to account for cost improvement across the organization. Both cost improvement and the traditional learning curve are defined by:

$$Y = AX^b$$

Where:

Y	=	Hours/unit (or constant dollars per unit)
A	=	First unit hours (or constant dollars per unit)
X	=	Unit number
b	=	Slope of the curve related to learning.

(There are two interpretations concerning how to apply this equation. In the unit interpretation, Y is the hours or cost of unit X only. In the cumulative average interpretation, Y is the average hours or cost of all units from 1 to X, inclusive.)

In parametric models, the learning curve is often used to analyze the direct cost of successively manufactured units. Direct cost equals the cost of both touch labor and direct materials in fixed year dollars. This is sometimes called an improvement curve. The slope is calculated using hours or constant year dollars. Chapters 3, Cost Estimating Relationships, and 10, Technical Evaluation of Parametrics, present a more detailed explanation of improvement curve theory.

Production Rate

Many innovations have been made in cost improvement curve theory. One is the addition of a variable to the equation to capture the organization's production rate. The production rate is defined as the number of items produced over a given time period. This equation modifies the basic cost improvement formula to capture changes in the production rate (Q^r) and organizational cost improvement (X^b):

$$Y = AX^bQ^r$$

Where:

Y	=	Hours/unit (or constant dollars per unit)
A	=	First unit hours (or constant dollars per unit)
X	=	Unit number
b	=	Slope of the curve related to learning
Q	=	Production rate (quantity produced during the period)
r	=	Slope of the curve related to the production rate.

The equation is generally applicable only when there is substantial production at various rates. The production rate variable (Q^r) adjusts the first unit dollars (A) for various production rates during the life of the production effort. The equation also yields a rate-affected slope related to learning. Chapter 10 provides additional information on data adjustments for inflation, learning, and production rate.

Significant Data Normalization Adjustments

The section describes some of the more complex adjustments analysts make to the historical cost data used in parametric analysis.

Adjustment for Consistent Scope

Adjustments are necessary to correct for differences in program or product scope between the historical data and the estimate being made. For example, suppose the systems engineering department compared five similar programs, and found that two included design-to-cost (DTC) requirements. To normalize the data, the DTC hours must be deleted from those two programs to create a data set with consistent program scope.

Adjustment for Anomalies

Historical cost data should be adjusted for anomalies (unusual events) when it is not reasonable to expect the new project estimates to contain these unusual costs. The adjustments and judgments used in preparing the historical data for analysis should be fully documented. For example, development test program data are collected from five similar programs, and it is noted that one program experienced a major test failure (e.g., qualification, ground test, flight test). A considerable amount of labor resources were required to fact-find, determine the root cause of the failure, and develop an action plan for a solution. Should the hours for this program be included in the database or not? This is the kind of issue analysts must consider and resolve. If an adjustment is made to this data point, then the analyst must thoroughly document the actions taken to identify the anomalous hours. There are other changes for which data can be adjusted, such as changes in technology. In certain applications, particularly if a commercial model is used, the model inputs could be adjusted to account for improved technologies (see the discussion of commercial models in Chapter 3). In addition, some contractors, instead of normalizing the data for technology changes, may deduct estimated savings from the bottom-line estimate. Any adjustments made by the analyst to account for a technology change in the data must be adequately documented and disclosed.

For instance, suppose electronic circuitry was originally designed with discrete components, but now the electronics are a more advanced technology. Or, a hardware enclosure which was made from aluminum is now made, due to weight constraints, of magnesium -- what is the impact on production hours? Perfect historical data may not exist, but good judgment and analysis by an experienced analyst should supply reasonable results.
Suppose the analyst has collected four production lots of manufacturing hours data:

Lot	Total Hours	Units	Average hours per unit
Lot 1	256,000	300	853 hours/unit
Lot 2	332,000	450	738 hours/unit
Lot 3	361,760	380	952 hours/unit
Lot 4	207,000	300	690 hours/unit

Clearly, Lot 3's history should be investigated since the average hours per unit appear high. It is not acceptable, though, to merely "throw out" Lot 3 and work with the other three lots. A careful analysis should be performed on the data to determine why it exhibits this behavior.

Data Adjustment Analysis Example

Suppose the information in the table represents a company’s historical data, and that the planned system is similar to one built several years ago.

Parameter	Historical System	Planned System
Date of Fabrication	Jul 98-Jun 00	Jul 03-Dec 03
Production Quantity	500	750
Size - Weight	22 lb. external case 5 lb. int. chassis	20 lb. external case 5 lb. int. chassis

	8lb. elec. parts	10 lb. elec. parts
Volume	1 cu ft-roughly cubical 12.l x 11.5 x 12.5	.75 cu ft-rec. solid 8 x 10 x 16.2
Other Prog Features	5% elec. Additional spare parts	5% elec. No spare parts

These data need several adjustments. In this example, the inflation factors, the difference in production quantity, the rate of production effect, and the added elements in the original program (spare parts) all require adjustment. The analyst must be careful when normalizing the data. General inflation factors are usually not appropriate for most situations; ideally, the analyst will have a good index of costs specific to the industry and will use labor cost adjustments specific to the company. The quantity and rate adjustments must consider the effects of quantity changes on the company's vendors and the ratio of overhead and setup to the total production cost. Likewise, for rate factors each labor element will have to be examined to determine how strongly the rate affects labor costs. On the other hand, the physical parameters do not require significant adjustments.

The first order normalization of the historic data would consist of:

- Material escalation using industry or company material cost history.
- Labor escalation using company history.
- Material quantity price breaks using company history.
- Possible production rate effects on touch labor (if any) and unit overhead costs.

Because both cases are single lot batches, and are within a factor of two in quantity, only a small learning curve adjustment would be required. Given the schedule shown, a significant production rate adjustment is needed.

Evaluation Issues

DFARS 215-407-5, "Estimating Systems," states that "contractors should use historical data whenever appropriate...." and that, "a contractor's estimating system should provide for the identification of source data and the estimating methods and rationale used to develop an estimate." Therefore, all data, including any adjustments made, should be thoroughly documented by a contractor so that a complete trail is available for verification purposes. Some key questions evaluators may ask during their review of data collection and analysis processes include:

- Are sufficient data available to adequately develop parametric techniques?
- Has the contractor established a methodology to obtain, on a routine basis, relevant data on completed projects?
- Are cost, technical, and program data collected in a consistent format?
- Will data be accumulated in a manner that will be consistent with the contractor's estimating practices?
- Are procedures established to identify and examine any data anomalies?
- Were the source data used as is, or did they require adjustment?
- Are adjustments made to the data points adequately documented to demonstrate that they are logical, reasonable, and defensible?

Chapter 7, Government Compliance, provides additional information on Government evaluation criteria.

Other Considerations

Several other issues should be considered when performing data collection and analysis.

Resources

Data collection and analysis activities require that companies establish sufficient resources to perform them, as well as formal processes describing data collection and analysis. Chapter 7, Regulatory Compliance, provides information on estimating system requirements, and discusses data collection and analysis procedures.

Information in the Wrong Format

While the contractor may indeed possess a great deal of data, in many cases the data is not in an appropriate format to support the parametric techniques being used. For example, commercial parametric models may have a unique classification system for cost accounts that differ from those used by a company. As a result, companies using these models would have to develop a process that compares their accounting classifications to those used by the model (also known as "mapping"). In other situations, legacy systems may generate data, to meet the needs for reporting against organizational objectives, which do not directly translate into the content or format needed for cost estimating and analysis. For example, many past and existing information systems have focused on the input side with little or no provision for making meaningful translations of output data for CER development or similar types of analysis. The growing use of ERP systems, which have a common enterprise-wide database, should improve this situation. Most large organizations are implementing ERP systems, or are reengineering their existing information systems, so that parametric estimating models can easily interface with them.

Differences in Definitions of Categories

Many problems occur when the analyst or the database fails to account for differences in the definitions of the WBS elements across the projects it contains. Problems also occur when the definitions of the contents of cost categories fail to correspond to the definitions of analogous categories in existing databases. For example, some analysts put engineering drawings into the data category while others put engineering drawings into the engineering category. A properly defined WBS product tree and dictionary can avoid or minimize these inconsistencies.

The Influence of Temporal Factors

Historical data are generated over time. This means that numerous dynamic factors will influence data being collected in certain areas. For example, the definition of the content of various cost categories being used to accumulate the historical data may change as a system evolves. Similarly, inflation changes will occur and be reflected in the cost data being collected over time. As DoD deals with a rapidly changing technical environment, both cost and non-cost data generated for a given era or class of technology can quickly become obsolete. Many analysts therefore consider a data-gathering project a success if they obtain five to ten good data points for certain types of hardware.

Comparability Problems

Comparability problems include, but are not limited to, changes in a company's department numbers, accounting systems, and disclosure statements. They also include changing personnel from indirect to direct charge for a given function. When developing a database, the analyst must normalize it to ensure the data are comparable. For example, when building a cost database, the analyst must remove the effects of inflation so that all costs are displayed in constant dollars.

The analyst must also normalize data for consistency in content. Normalization for content ensures that a particular cost category has the same definition in terms of content for all observations in the database. Normalizing cost data is a challenging problem, but it must be resolved if a good database is to be constructed.

Database Requirements

Resolving database problems to meet user needs is not easy. For example, cost analysis methodologies may vary considerably from one analysis or estimate to another, and the data and information requirements for CERs may not be constant over time. An analyst's data needs now do not determine all future needs, and must be periodically reviewed.

The routine maintenance and associated expense of updating the database must also be considered. An outdated database is of little use in forecasting future acquisition costs. The more an organization develops and relies on parametric estimating methods, the more it needs to invest in data collection and analysis activities. The contractor must balance this investment against the efficiency gains it plans to achieve through use of parametric estimating techniques. If the contractor moves towards an ERP system, the incremental cost to add a parametric estimating capability may not be significant.

Good data underpins the quality of any estimating system or method. As the acquisition community moves toward estimating methods that increase their reliance on the historical costs of the contractor, the quality of the data cannot be taken for granted. Industry and their Government customers should find methods to establish credible databases that are relevant to the history of the contractor. From this, the contractor will be in a better position to reliably predict future costs, and the Government to evaluate proposals based on parametric techniques.

III. CER DEVELOPMENT

INTRODUCTION

In estimating the probable cost of a proposed new system, the basis of parametric cost estimating is the philosophy of "what's past is prologue." The structural foundation of modern cost analysis is the cost-estimating relationship (CER), statistically derived from historical cost data. CERs are usually expressed in the form of linear or curvilinear statistical regression equations that predict cost (the "dependent" variable) as a function of one or more "cost drivers" ("independent" variables). However, the CER tells only part of the story. At the beginning of a cost-estimating task lies the work-breakdown structure (WBS), a list of everything that has to be paid for in order to bring a system to its full operational capability. A WBS includes "high-level" categories such as research, development, and testing; production; launch (if applicable); and operations, maintenance, and support. At "lower" levels of the WBS, costs of software modules, electronic boxes and other components are accounted for. A CER can be established for any WBS element at any level, as long as data are available to support its derivation. It should be pointed out, though, that in addition to CERs, the cost estimator uses analogies to specific items already developed and produced, vendor price quotes for off-the-shelf items, or other appropriate techniques to assign a dollar value of estimated cost to each item listed in the WBS. This section, however, focuses solely on the development of CERs, the primary development tool of which is statistical regression.

Simply stated, a CER is a mathematical expression, most commonly an algebraic function, that describes the cost of a system, subsystem, component, or activity in terms of numerical values of one or more "cost drivers." The cost drivers are the independent variables of the functional relationship, and the cost being described is the dependent variable. The values of the dependent variable are determined, or "driven," by the values of the independent variables. The CER has a great advantage over other estimating methods in its ability to assess sensitivity of the cost to variations in the cost drivers. This can be done through an application of (usually) simple calculus. Those analysts who are familiar with the derivative and its meaning will realize that the partial derivative of cost with respect to any one cost driver expresses the rate of change of cost value with respect to changes in that cost-driver value. This means that, applying the derivatives of the CER, we can say that cost rises (for example) $2,000 for each pound of increase in weight of the avionics or $650 for each additional line of code in the command and control software CSCI.

Cost-to-cost CERs are CERs of a special kind in which cost of a system, subsystem, component, or activity is the dependent variable, while costs of other items are the independent variables. Examples include (but are not restricted to) system engineering and program management costs, which are often expressed as a percentage of hardware and software costs, quality assurance costs, which are sometimes expressed as a percentage of manufacturing costs, and development costs, which are occasionally expressed as a percentage of production costs. In general, the cost of one element is used to estimate, or predict, that of another.

The more common kind of CER expresses the cost of an item (system, subsystem, component, or activity) in terms of some intrinsic characteristics of the item. For example, CERs express the cost (dependent variable) of an item in terms of independent variables such as its weight, the number engineering drawings required for its design, some measure of its heritage from previously produced items of the same kind, or the difference between its current technology-readiness level (TRL) and the TRL required for its final production and deployment.

Occasionally, it may be appropriate to use "yes or no" discrete cost drivers, such as whether a satellite is spin-stabilized or three-axis stabilized or whether an airplane is land-based or carrier-based. To turn "yes or no" into numerical terms so that cost driver can be part of a statistically-derived CER, we can indicate "yes" by the number 1 and "no" by the number 0. Such a cost driver typically enters the CER in the form of an independent variable that is the exponent of a coefficient c of the CER, say c^w: for items in the satellite data base, for example, we define $w = 0$ for spin-stabilized and $w = 1$ for three-axis stabilized satellites, and for items in the airplane data base, we define $w = 0$ for land-based and $w = 1$ for carrier-based airplanes. The coefficient c is one of those solved for when the CER is statistically derived from the supporting data base.

It is important to note that use of the term "cost driver" to describe the independent variables in a CER does not necessarily imply that variations in the cost driver actually cause changes in cost. In many situations cost drivers, such as weight, represent a conglomeration of other factors, such as complexity, redundancy, multimission capability, etc., that themselves cause cost changes but are difficult individually to quantify explicitly. However, when weight is substituted for these "real" cost drivers, it may be correlated with (or "track") cost well enough, so that if the weight is known, then the cost can be estimated fairly well. In other cases, of course, weight may be an inappropriate representative of the real cost drivers, particularly in cases where smallness or lightness considerations are primary, such as in the case of electronic devices. Both sides of the argument also apply as well to other "surrogate" cost drivers, such as solar-array area, software lines of code, or data-processing throughput.

Regardless of the underlying cause and effect relationships, if there are any, in the context of this section cost drivers are assumed to be either true drivers of cost or valid surrogates for true cost drivers of the item being estimated.

CERs can also be used to estimate labor hours, developer-months, and other cost-like metrics, instead of raw dollar-valued costs. In addition, a related concept called the "time-estimating relationship" (TER) expresses project schedule duration in terms of "schedule drivers." TER-development theory in the abstract is indistinguishable from CER-development theory.

In general a CER is a valuable estimating tool that can be used at any time in the estimating process. For example, CERs may be used in the program concept or validation phase to estimate costs when there is insufficient system definition for more detailed approaches, such as the classical "grass roots" or "bottom-up" methods. CERs can also be used in a later phase of a program as primary estimates or as crosschecks of results based on non-parametric estimating methods, such as analogies or activity-based costing. CERs may also form the primary Basis of Estimate for proposals submitted to the Government or higher-tier contractors. They are also used extensively by government agencies to develop Independent Cost Estimates for major elements of future programs.

Classical least-squares regression (often referred to as "ordinary least squares" or "OLS"), developed by mathematicians in the 18th Century, has in the past been the statistical procedure of choice in deriving CERs. However, OLS has always imposed severe restrictions on an analyst who wants to derive functional relationships between dependent y and independent x variables, forcing him or her to model the error as additive when the relationship is linear ($\boldsymbol{y = a+bx}$), logarithmic ($\boldsymbol{y = a + b \log x}$), or polynomialic, but as multiplicative when the relationship is exponential ($\boldsymbol{y = ax^b}$) or power ($\boldsymbol{y = ab^x}$). This severely restricts his or her ability to optimally model cost-related phenomena. "General-error regression," introduced a decade ago to circumvent these problems, takes advantage of modern computing capability and advanced numerical analysis techniques, thereby offering the analyst the choice of optimizing the CER's quality metrics by minimizing additive or multiplicative error, regardless of the functional form of the relationship. In this section, we will discuss the merits and deficiencies of all these CER-development techniques.

A. QUALITY METRICS OF A COST-ESTIMATING RELATIONSHIP

In order to derive a CER for, say, a satellite's solar array panels, cost and technical data on solar arrays that have already been produced must be collected and organized into a consistent data base. For each array in the data base, cost is lined up against numerical values of appropriate cost-driving technical characteristics such as weight, area, and storage capacity. Various algebraic relationships between cost and one or more possible cost drivers that make both engineering and statistical sense are then compared to find out which relationship is the "optimal" predictor of solar-array cost. It is important to understand that which criteria are the best ones to use for assessing CER optimality is neither *a priori* obvious nor derivable mathematically nor a subject on which all cost analysts agree. Three statistical criteria, however, lie at or near the top of almost everyone's list: the sample standard error of estimates made by the CER of points in the data base, the sample bias of estimates of the points in the data base, and the correlation between the actual costs in the data base and the CER-based predictions of those costs. Descriptors of regression-function quality that require almost fortuitous juxtapositions of special statistical circumstances (e.g., statistical independence, linearity, Gaussian residuals, homoscedasticity, and non-multicollinearity) whose existence in the cost-analysis context can rarely, if ever, be convincingly established are less valuable as quality metrics because their applicability generally cannot be validated. Included among these descriptors are the classical ***t*** and ***F*** scores, analysis-of-variance (ANOVA) tables, and alpha levels for statistical tests of significance.

Formal definitions of the three primary measures of CER statistical quality may be given as follows, for additive-error and multiplicative-error CERs, respectively:

1. The sample standard error of the estimate is a "one-sigma" number that can be used to bound actual cost within an interval surrounding the estimate with some degree of confidence:

> <u>Sample Standard Error (Additive-Error CERs)</u>: Root-mean-square (RMS) of all additive errors (i.e., differences between estimate and actual) made in estimating points of the data base using the CER, normalized by the number of data points and CER coefficients;
> <u>Sample Standard Error (Multiplicative-Error CERs)</u>: Root-mean-square (RMS) of all percentage errors (i.e., differences between estimate and actual,divided by the estimate) made in estimating points of the data base using the CER, normalized by the number of data points and CER coefficients;

2. The sample bias of the estimate is a measure of how well overestimates and underestimates of data-base actuals are balanced:

> <u>Sample Additive Bias</u>: Algebraic sum, including positives and negatives, of all additive errors (i.e., differences between estimate and actual) made in estimating points of the data base using the CER, divided by the number of data points;
> <u>Sample Percentage Bias</u>: Algebraic sum, including positives and negatives, of all percentage errors (i.e., differences between estimate and actual,divided by the estimate)

made in estimating points of the data base using the CER, divided by the number of data points;

3. The sample correlation-squared is the R-squared value of the relationship between estimates and their corresponding actuals, which ideally should be the straight line $y = x$.

Sample Correlation-Squared between Estimates and Actuals: If the CER were a perfect predictor of the actual cost values of elements of the data base, a plot of estimates against the actuals would follow the 45° line $y = x$ quite closely (correlation-squared, or R^2, a statistical measure of the extent of linearity in a relationship between two quantities, will be high if estimates do indeed track actuals but low if they do not).

1. ORDINARY LEAST SQUARES LINEAR REGRESSION

Suppose we have a set of pairs of data points, $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n),$ where each x value represents a numerical value of a technical or performance parameter (such as weight, power, or thrust, for example) that is a cost driver, while the corresponding y value represents the cost of a system, subsystem, or component whose cost driver is x. It is desired to establish a mathematical relationship $y = f(x)$ between x and y so that cost can be estimated on the basis of the cost driver. Unfortunately, it is almost always the case that the points do not lie exactly along a straight line or simple curve $y = f(x)$. Furthermore there is no single "best-fitting" line or curve, because the meaning of the term "best fitting" is not universally agreed up. There is, instead, a wide range of lines or curves from among which the cost analyst must make his or her selection in order to satisfy a set of preferred optimization criteria.

Ordinary least squares (OLS), the most popular theory of curve fitting for over three centuries, is well-known to anyone acquainted with elementary statistics. The reasons for the popularity of OLS are the simplicity of its optimization criterion and the fact that it was amenable to hand computation in the pre-computer age. Yet, OLS is only one of many techniques that can be used to fit lines or curves. Nevertheless, any discussion CER development must begin with OLS.

OLS theory asserts that $y = a + bx + \varepsilon$, where a and b are constants derived from the historical data and ε is a Gaussian (i.e., normally distributed) mean-zero random error term independent of the cost-driver value x ("homoscedasticity"). The regression model $y = a + bx + \varepsilon$ is an "additive-error" model, where the error of estimation ε is expressed in the same kind of units as is y, namely dollars. Furthermore, the Gaussian assumption on ε implies that, given x, y is a normally distributed random variable with mean $a+bx$ and variance σ^2 equal to that of ε. Given $x = x_i$, the historical data point y_i is then a realization of a Gaussian random variable with mean $a+bx_i$ and variance σ^2. For each i, the difference (i.e., residual) $y_i-(a+bx_i) = y_i-a-bx_i$ between the observed and expected values of y is an indicator of how well the straight line $y = a+bx$ "fits" the data. According to OLS theory, the constants a and b are chosen so that the sum of squared differences ("additive errors")

$$E(a,b) = \sum_{i=1}^{n} \left(y_i - a - bx_i\right)^2$$

is as small as possible. This criterion is the origin of the term "least squares." See Figure 5 for an illustration.

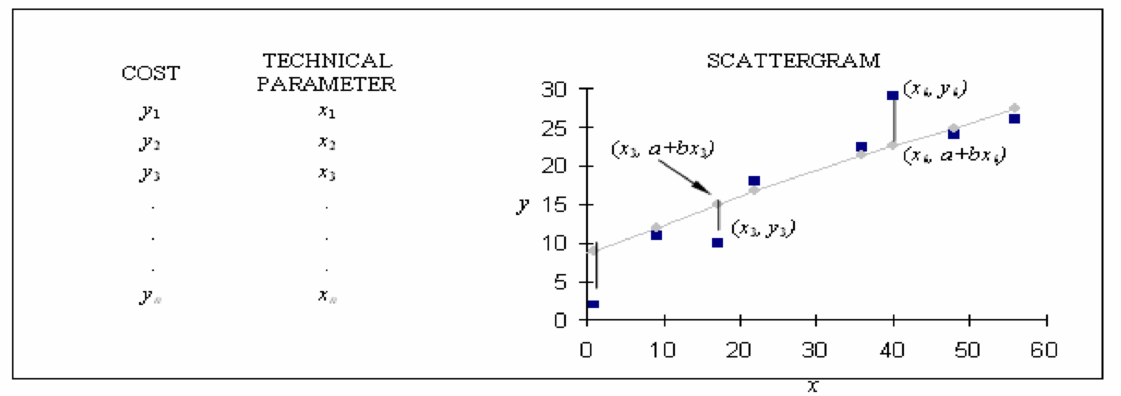


Figure 5: Historical Data Fitted to *y = a+bx*.

The attractiveness of OLS derives in large part from the fact that explicit expressions for ***a*** and ***b*** can be found by solving a pair of simultaneous linear equations.

There are several ways to measure the quality of an OLS CER, ranging from statistical ***t*** tests of the significance of the coefficients ***a*** and ***b*** to F tests of significance of the R^2 value to the sample standard error and bias of the CER itself. Details of those statistical tests, which work best under conditions of independent normally distributed residuals (i.e., differences between estimates and actuals) and homoscedasticity, can be found in elementary statistics textbooks of the kind you may have used in college. In this section, we will focus on the three quality metrics defined above. The sample standard error of the estimate, expressed in dollars for an OLS CER is, a multiple of the square root of the expression ***E(a,b)*** above that is minimized in order to determine the coefficients ***a*** and ***b*** of the straight-line CER. This means that the OLS CER is the linear relationship between cost and cost driver that has the smallest possible sample standard error of the estimate.

The sample bias, also expressed in dollars, is the mean of the residuals $(a+bx_i)-y_i$, but does not have to be calculated by the analyst, since its numerical value always works out to be zero for an OLS CER.

The correlation-squared, denoted R^2, is a unitless quantity that, in the case of OLS linear CERs only, represents both (1) the percentage of variation in actual cost that is attributable to variations in the value of the cost driver and (2) the percentage of variation in estimated costs that is attributable to variations in the corresponding actual costs. The algebraic formula for R^2, in the OLS case always (if calculated correctly) results in a number between 0 and 1 and, when converted to a percentage, between 0% and 100%.

EXAMPLE: AN OLS LINEAR CER

Consider the following set of cost and technical data, where ***n*** represents the number of data points in the data base supporting the CER, ***x*** represents the diameter in feet of a ground antenna and ***y*** represents is cost in thousands of FY07 dollars.

n	*x*	*y*
7	7.9	5.900
	8.2	13.300
	9.8	43.595
	11.5	72.944
	16.4	95.434
	19.7	106.074
	23.6	116.274

Table 1. Cost/Technical Data Base (*x* = technical parameter, *y* = cost)

We would like to derive a CER of the linear algebraic form ***y = a + bx*** to model the data base and to apply to future estimating situations. If we choose to derive an OLS linear CER, we can apply the formulas that were established for that purpose. It should be noted that these formulas are built into Microsoft's Excel® program, so the computation of ***a*** and ***b*** can be immediate for those analysts knowledgeable in the intricacies of Excel®. No matter how the derivation is done, the results are the following:

b =	6.797
a =	-29.499
r =	0.934
R^2 =	87.1589%
Std Error =	17.503
Bias =	0.000

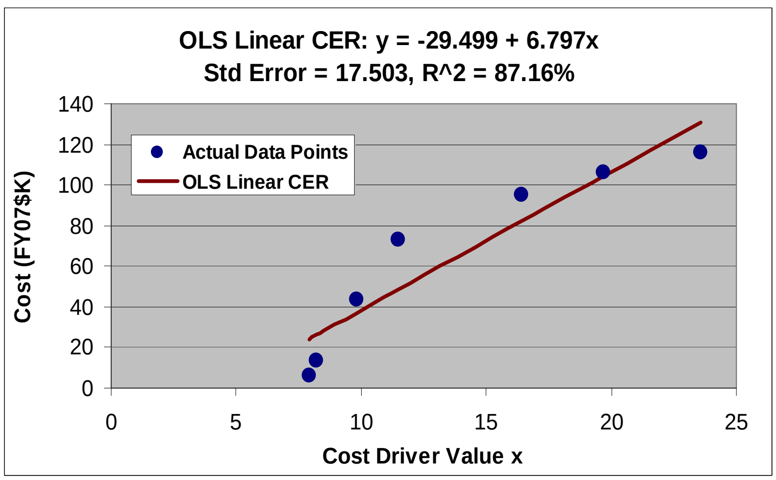


Figure 6: OLS Linear CER and its Quality Metrics.

You will notice that we have superimposed a linear model on a set of data that is obviously nonlinear. We have done this with two specific instructional purposes in mind: (1) to demonstrate that a CER of ***any*** algebraic form can be imposed on ***any*** set of data; and (2) to demonstrate that the validity of the statistical modeling process can be assessed using the quality metrics, as well as the graphical comparison above.

The OLS linear approach to CER development has both advantages and disadvantages. Its primary advantage is the fact that, having been developed in the 18th century and publicized since then in almost all venues where statistics is taught and studied, it is very well known by almost all cost analysts and its processes are fully understood by most. Another major advantage is that its implementation is very simple, as the required formulas have been built into not only Excel®, as noted above, but into all other COTS and GOTS statistical and cost-estimating software products.

On the other hand, OLS also has a number of characteristics that are disadvantages from the cost estimator's point of view. One disadvantage of the OLS linear approach to CER development is that the accuracy of the statistical inferences associated with it, namely the $\boldsymbol{t}$ and $\boldsymbol{F}$ tests for the validity of the coefficients $\boldsymbol{a}$ and $\boldsymbol{b}$ and the $\boldsymbol{R^2}$ value, is heavily dependent on the required underlying conditions, namely independent normally distributed residuals and homoscedasticity, being satisfied. Due to the common paucity of data upon which CERs are based, it is almost impossible for these underlying conditions to be validated. However, the primary disadvantage using OLS to derive linear CERs is that it is an "additive-error" model, i.e., the standard error of the estimate is expressed in dollars, <u>not</u> as a percentage of the estimate. Many cost estimators have been asked, after delivering their estimates, "How accurate is your estimate – to within what percent?" Answering this question can be a problem for a cost analyst who has used an OLS-derived CER. Consider, for example, the CER in Figure 2, whose standard error is $17,503. If the value of the cost driver is around 20 or 25, there is no problem as the estimate is around $110,000, so the one-sigma error is about 16% of the estimate. However, if the value of the cost driver is around 7 or 8, the estimate itself is only around $25,000, so one-sigma error of $17,503 represents a possible error of around 70%. Not so impressive, is it?

WHY PERCENTAGE ERROR?

While it is true that the computational simplicity of OLS is impressive, the fact noted above that it requires an additive-error model reduces its value to cost analysts. A multiplicative-error model, in which both the standard error of the estimate and the bias are expressed as percentages of the estimate, is more meaningful in the cost-estimating context. Two practical benefits of the multiplicative-error model accrue to the estimator.

First, expressing cost-estimating error in percentage terms offers stability of meaning across a wide range of programs, time periods, and estimating situations. A percentage error of, say 40%, retains its meaning whether a $10,000 component or a $10,000,000,000 program is being estimated. A standard error expressed in dollars, say $59,425, is an extremely huge error when estimating a $10,000 component, but is even less significant than a typographical misprint when reported in connection with a $10,000,000,000 program. Even in cases that are not so extreme, a standard error expressed in dollars quite often makes a CER virtually unusable at the low end of its data range, where relative magnitudes of the estimate and its standard error are inconsistent.

Second, in the case of bias, a dollar-valued expression (the algebraic sum, including positives and negatives, of dollar-valued errors made in estimating points of the data base) would not be as informative as an expression of bias in terms of percentage of the estimate, because a particular amount of dollars of bias would not have the same impact at all points of the data base. See Figure 7 for an illustration of the difference between the error patterns of additive-error and multiplicative-error CERs.

Furthermore, use of OLS additive-error regression sometimes produces a curve fit that is biased toward high-cost programs, namely those data points with the larger observed values. This is due to the fact that the additive-error model attempts to minimize the sum of squared deviations from all data points, thus giving the larger data points a perhaps unduly large influence in determining the "best-fitting" curve. Use of the percentage-error model will reduce the influence of the large data values, because it will be the percentage errors in <u>all</u> estimates that will be reduced to their minima. It is therefore useful to have a general least-squares methodology that can treat not only additive, but also percentage-error models.

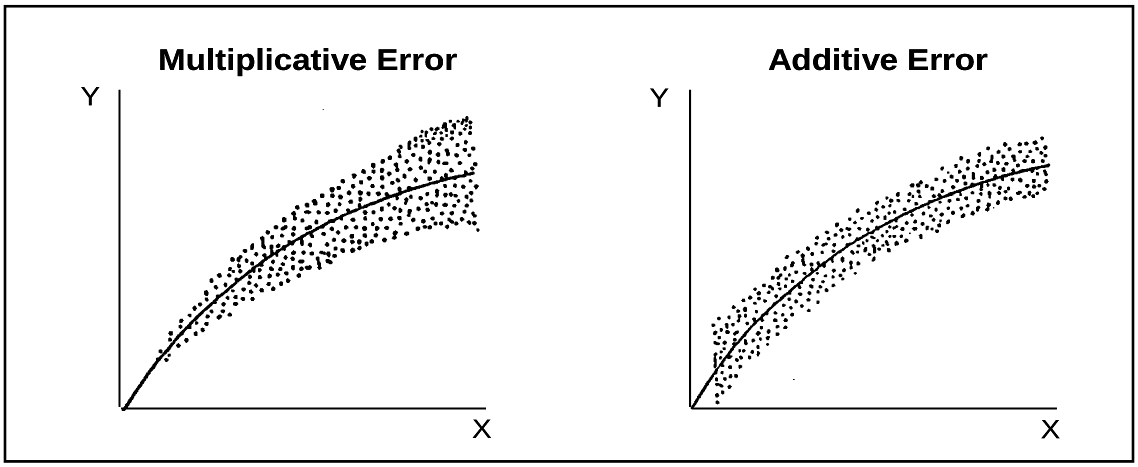


Figure 7: Error Patterns Surrounding the CER, Based on Percentage Error and Additive Error, Respectively. (H..L. Eskew and K.S. Lawler, Reference 7, page 107).

2. THE LOGARITHMIC-TRANSFORMATION TECHNIQUE

Development of <u>nonlinear</u> CERs in most cost models involving historical data has in the past been based largely on explicit solutions of the classical least-squares linear regression equation $Y = A + bX + E$, where Y is the logarithm of the cost y, X is the logarithm of the numerical value x of a cost driver, E is a Gaussian error term whose variance does not depend on the numerical value of X, and A and b are numerical coefficients derived from the historical data. The coefficients of nonlinear CERs such as the most common "power" CER form $y = ax^b\varepsilon$ have, since the 18th Century, been derived by taking logarithms of both sides and reducing the formulation to the linear form $log(y) = log(a) + blog(x) + log(\varepsilon)$, which is equivalent to the expression above if we denote $Y = log(y)$, $X = log(X)$, $A = log(a)$ and $E = log(\varepsilon)$. The symbol "**log**" can represent either the common (base 10) or natural (base e) logarithm or, for that matter, a logarithm of any base whatsoever, as long as it retains the same meaning throughout a complete sequence of computations. The explicit OLS expressions for a and b in nthe linear model can then be applied to calculate the numerical values of $A = log(a)$ and b, respectively. We calculate a using the formula $a = 10^A$ if common logarithms are being used and $a = e^A$ in the case of natural logarithms. By analogy, the standard error of the estimate is output by most statistical software packages according to the same formulas as those for OLS linear CERs, except that the numerical values of x and y are replaced by their logarithms. This means that the standard error of the estimate and bias are optimized to maximize the accuracy with which ***log x*** can be used to predict ***log y***, not the accuracy with which ***x*** can be used to predict ***y***. The value of R^2, furthermore, measures the quality of the linear fit of ***log y*** to ***log x***, not the nonlinear fit of ***y*** to ***x***.

EXAMPLE: AN OLS-DERIVED NONLINEAR CER

Using the same data set as before, but applying the logarithmic transformation before exercising the computational formulas, we make the required calculations and

n	x	y	X = log(x)	Y = log(y)
7	7.9	5.900	0.898	0.771
	8.2	13.300	0.914	1.124
	9.8	43.595	0.991	1.639
	11.5	72.944	1.061	1.863
	16.4	95.434	1.215	1.980
	19.7	106.074	1.294	2.026
	23.6	116.274	1.373	2.065

Table 2. Data Base Set Up for Deriving OLS Logarithmic-Transformed "CER".

obtain the following coefficients and quality measures for the log-log linear "CER" $\log y = A + b \log x$ and its back-transformed power CER $y = ax^b$:

b =	2.304
A =	-0.911
r =	0.866
R^2 =	74.9673%
Std Error =	0.276
Bias =	0.000

b =	2.304
a =	0.123
R^2 =	75.5173%
Std Error =	71.7553%
Bias =	-14.7891%

Table 3. Coefficients and Quality Statistics of
OLS Logarithmic-Transformed "CER" (at left) and its Corresponding Power CER (at right).

The following graph displays the power CER superimposed on the data set:

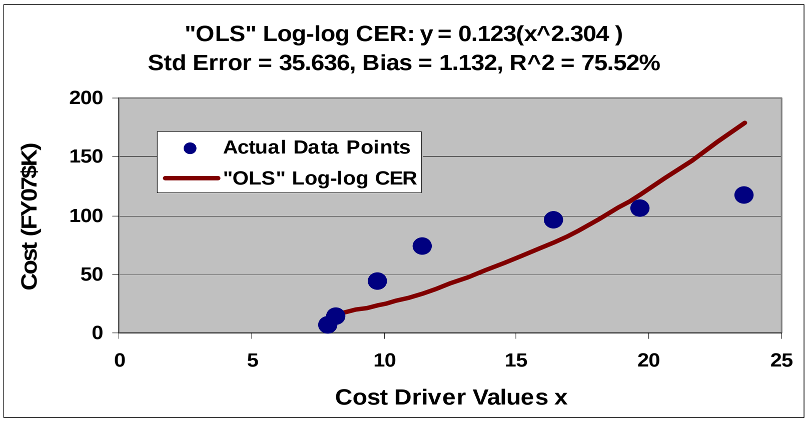


Figure 8: OLS-Derived Power CER Superimposed on the Data Base.

A few quick observations can be made on the basis of the material displayed in Table 3 and Figure 4. First of all, we notice from Table 3 that the bias of power CER is negative, namely -14.7891%. This means that a power CER that is derived by back-transforming an OLS-derived log-log "CER" on the average underestimates the costs of projects in its data base. This means that, if nothing is done to fix the problem, such a CER will probably underestimate the cost of a project to which it is being applied. We will discuss below methods that have been used to bring power CER bias back to zero.

Second, we notice from Figure 8 that the CER's convexity (the way it bends) is different from the convexity of the data base. If enough of us deliver CERs like this to individuals outside our profession, our reputations for common sense will be dealt a fatal blow! To avoid situations like this in the future, we must understand the reason it has occurred here. The power CER $y = ax^b$ has no "fixed-cost" or intercept term like the a in the linear CER $y = a + bx$. This means that a power CER must pass through the origin, namely the zero-zero point on its graph. You might remember from your study of regression in your statistics class that a regression line or curve does not necessarily pass through any of the points in its data base. That's true – a power CER does not have to pass through any of the real data points. But all power CERs do have to pass through the origin (which is usually not a data point), because a power CER has no intercept term. The fact that the power CER must pass through the origin is what accounts for the mismatch between the convexities of the CER and its supporting data base in Figure 8. We will discuss later how to fix this problem by introducing the so-called "triad" CER form, namely $y = a + bx^c$, which includes both the linear and the power CER forms as special cases, the linear when $c = 1$ and the power when $a = 0$.

3. ISSUES ASSOCIATED WITH THE LOGARITHMIC TRANSFORMATION

Unfortunately, the meanings of the quality metrics standard error, bias, and R^2, are, the case of nonlinear CERs derived by OLS methods, easy to misunderstand and, if fact, are almost always misrepresented. This situation is due to the fact that all computations are being made in "logarithmic space," i.e., we are working with the logarithms of the data points, rather than with the data points themselves. It is important that we discuss this issue in full detail, so that the correct meanings of the quality metrics of nonlinear CERs are understood, at least by readers of the *Parametric Estimating Handbook*.

First, the standard error of the estimate ***SEE*** is expressed in meaningless units ("log dollars"), just as the data-base values of ***log y*** are. It would be useful if this quantity, 0.276 in Table 3 above, were equivalent to a percentage of the estimate, namely 27.6%, as one would think the case to be in a multiplicative-error model such as $y = ax^b\varepsilon$. In fact, the power CER standard error has been incorrectly reported as such on many occasions**.** However, P.H. Young (Reference 13) shows that this ***SEE*** is not often equal or even close to the percentage error of the estimate, and he recommends that the standard percentage error be calculated directly in "arithmetic space," namely using the data points themselves, rather than their logarithms. To do this, we have to go back to the original data points and the "de-logged" CER of the form $y = ax^b$. The percentage standard error of the estimate will then have the formula

$$\%SEE = \sqrt{\frac{1}{n-2}\left(\frac{y_i - ax_i^b}{ax_i^b}\right)^2} \quad ,$$

which expresses the error as a ratio of actual residual to the actual estimate, without the interposition of logarithms. Notice that in Table 3, the correct standard error of the power CER is reported as 71.7553%, while the logarithmic standard error of 0.276 would, if converted to a percentage, indicate 27.6%.

Having said all this, we now have some more bad news to report. The CER $y = ax^b$, as calculated in logarithmic space, will not actually have the smallest possible ***%SEE***, i.e., it will not be the CER of that form that "best fits" the data points. The reason for this is that the numerical values of a and b were selected to minimize the standard error of the estimate in logarithmic space. But because, when we apply the CER, we will be doing our estimating in arithmetic space, i.e., the geometric space where the actual data points reside, we really should have selected our a and b values to minimize the ***%SEE*** quantity just above. Then we would have obtained the CER of the form $y = ax^b$ that has the smallest possible sample percentage error of the estimate. We will illustrate this process later in this section when we introduce the MPE-ZPB method.

The same can be said about the ***Bias*** value. Although the logarithmic-space CER $log(y) = log(a) + blog(x) + log(\varepsilon)$ is unbiased (inheriting that property from the OLS mathematics by which it was derived), the back-transformed arithmetic space CER $y = ax^b$, the one we plan to use for estimating, is usually biased low in the additive (dollar-valued) sense. Note the bias of -14.7891 among the quality metrics of Table 3. This fact was forcing estimators to unknowingly produce "low-ball" cost estimates on a consistent basis until the bias was recognized by the cost-estimating community and a correction term developed for it. The existence of the bias was known and understood in the statistical community at least since 1960 when J. Neyman and E. Scott (Reference 11) published a number of correction factors. The one that has taken hold among cost estimators is the so-called "Ping factor"

$$e^{\frac{1}{2}(\frac{n-2}{n})s^2},$$

where s is the standard error of the estimate in logarithmic space. For details of the Ping factor, brought to the attention of the cost-estimating community in the middle 1980s, see S.P. Hu and A.R. Sjovold (Reference 9). Additional remarks on the subject can be found in References 1 and 2.

To obtain a zero-additive-bias CER of the form $y = ax^b$, we simply multiply the CER (or the constant a) by the Ping factor. Unfortunately, the process is not as simple as it sounds. After we carry out the multiplication, we now have a new CER, which we can denote as $y = cx^b$, where c equals a times the Ping factor. This new CER now has new quality metrics, because the old ones no longer apply. In general, its standard error will be larger and its R^2 will be smaller, assuming that the original CER (before applying the Ping factor) had the smallest possible standard error (in logarithmic space) and the largest possible R^2 (in logarithmic space). The only improvement is in the bias, which is now zero (dollars). To report the CER's full suite of correct quality metrics, it is therefore necessary to recalculate the standard error and R^2 of the new, Ping-adjusted CER, though model developers rarely carry out this step.

Finally, there is another issue associated with R^2, beyond the fact that it is routinely calculated using $\log x$ and $\log y$ values, rather than x and y values. Unless the CER is an OLS CER of linear form, even R^2 calculated using the OLS formula no longer signifies the percentage of variation in actual cost that is attributable to variations in the value of the cost driver or the percentage of variation in estimated costs that is attributable to variations in the corresponding actual costs. See Reference 4 for an explanation of this problem. To obtain a meaningful version of R^2, we must default to the definition of R^2 provided in the "**QUALITY METRICS …**" section above, namely the correlation-squared between estimates and actuals. In general, to obtain the correct values of ***Bias*** and R^2, it remains true that the CER's computation must be carried out in arithmetic space, not logarithmic space.

A further, and more serious, weakness of the logarithmic transformation method, is that an analyst using OLS for both linear and nonlinear CERs, is forced to assume an additive-error (uniform dollar value across the entire range of the cost driver) model when historical data indicate a linear relationship between cost driver and cost, but a multiplicative-error (a percentage of the estimate) model when a nonlinear relationship is indicated by the data. This means that a model that contains both linear and nonlinear CERs for different WBS elements will express the estimating error in dollars when the CER is linear, but as a percentage of the estimate when the CER happens to be nonlinear. This kind of inconsistency makes it difficult to explain cost estimating to external parties and brings disrepute upon the cost-estimating community.

Finally, use of the OLS-based transformations *a priori* excludes from consideration certain potentially attractive nonlinear forms, such as the aforementioned triad form $y= a + bx^c$, because a logarithmic (or any other reasonable) transformation fails to reduce the least-squares problem to the OLS framework. The idea that a linear CER can have a nonzero constant term a, but a nonlinear one cannot and so must pass through the origin, is another notion that has the potential to subject cost estimators to ridicule.

4. THE GENERAL-ERROR MODEL

The general-error regression model has been designed to circumvent the inherent difficulties that OLS presents to the cost-estimating community. It allows the analyst to specify, given any set of historical-cost data, that a percentage-error model, rather than OLS or any other additive-error model, is to be used in deriving linear or nonlinear least-squares CERs . Should there be a good reason to prefer an additive-error model in a particular circumstance, however, the general-error model can accommodate it for both linear and nonlinear CERs. The primary benefit of general percentage-error regression, though, is its ability to correctly model nonlinear CERs of any algebraic form. See Reference 2 for further comments and details.

All known weaknesses of OLS can be circumvented by applying general-error regression, which allows the analyst to determine the optimal coefficients for any curve shape and to choose the error model independently of the CER shape. Because of our inability to solve simultaneous equations that are nonlinear with respect to the unknowns, the general nonlinear regression solution must be obtained by an iterative procedure. Several such procedures, most of them based on Newton's method from elementary calculus, have been proposed and discussed in the statistical literature over the years. The optimal (error-minimizing) solution is found by sequential computer search rather than by explicit solution of simultaneous equations, as in OLS. No one should find this unusual as OLS was developed in the 18th Century, when explicit formulas were necessary for calculating anything. Nowadays, with 100 years of numerical analysis (i.e., calculus by computer) and 50 years of machine-computation experience in the general population behind us, we should feel comfortable deriving CERs by mathematical optimization techniques such as those available via the Excel® Solver routine, rather than by explicit formulas that we can work out using a hand calculator or even a slide rule.

Once we leave the domain of OLS, as noted above, there do not exist explicit formulas for CER coefficients that minimize the percentage standard error of the estimate or optimize other CER quality metrics. General-error regression is instead implemented in a number of commercial software packages, including Microsoft's Excel spreadsheet using the Excel® Solver routine, which handles complex nonlinear problems by building a worksheet with multiple changing cells. Suppose, for example, the analyst wants to fit a set of historical cost data to the nonlinear CER triad form $\boldsymbol{y = a + bx^c}$. In order to apply least-squares theory in the multiplicative-error environment, the analyst has to find the numerical values of $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ that minimize the sum of squared percentage errors of estimating each data point in the historical data base. The multiplicative-error CER is

$$\boldsymbol{y = (a + bx^c)\varepsilon},$$

where $\boldsymbol{\varepsilon}$ is the error. While the ideal magnitude of an additive-error term is 0 (which indicates that estimated cost exactly equals actual cost), the ideal magnitude of a multiplicative-error terms is 1 (which indicates in the multiplicative-error context that estimated cost exactly equals actual cost). The error term is then equal to

$$\boldsymbol{\varepsilon = \frac{y}{a + bx^c}}.$$

We want to choose values of $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$ that minimize the sum of squared percentage errors of estimating each data point:

$$SS\%E = \sum_{i=1}^{n} (\varepsilon_i - 1)^2 = \sum_{i=1}^{n} \left(\frac{y_i}{a + bx_i^c} - 1 \right)^2 = \sum_{i=1}^{n} \left(\frac{y_i - a - bx_i^c}{a + bx_i^c} \right)^2 .$$

Using Excel® Solver or some other appropriate mathematical tool, we can in fact find values of ***a***, ***b***, and ***c*** that nearly minimize SS%E. (With any implementation of a numerical analysis technique, especially one included free with a general-usage computer program as Excel® Solver is, we can never be sure of attaining the exact minimum point, because complicated multivariate mathematical expressions have too many hills and valleys to be tested – but we can get fairly close in most cases.) However, it turns out that percentage bias of the resulting CER is positive, not zero. There are two different excursions of this technique that can be used to derive percentage-error CERs having zero bias: (1) iteratively reweighted least squares (IRLS) and (2) MPE-ZPB constrained optimization, i.e. minimizing standard percentage error subject to the constraint that the percentage bias be zero. We will discuss each of these excursions in turn.

5. ITERATIVELY REWEIGHTED LEAST SQUARES

IRLS was originally proposed in a 1968 article by J.A. Nelder (Reference 10) as a method of deriving multiplicative-error "generalized linear" CERs having zero percentage bias. The theory of IRLS was advanced significantly to CERs of arbitrary algebraic form in a tradition-breaking article by R.W.M. Wedderburn (Reference 14) six years later. Wedderburn, however, did not claim in his thoroughgoing 1974 article that IRLS CERs actually minimize the percentage error of the estimate. Rather, Wedderburn demonstrated that IRLS maximized a construct called the "quasi-likelihood," which is related to the statistical likelihood function and, in fact, equal to it in the special case when the standard error of the estimate is proportional to the estimate (as in the percentage-error model) and the random error has a gamma distribution. Ordinary least squares (OLS), it should be noted, maximizes the "likelihood" function under its classical conditions of normally distributed additive errors with constant variance across the range of data ("homoscedasticity").

It is important to note that IRLS CERs populate the Air Force's Unmanned Space Vehicle Cost Model, Version 7 (Reference 12) and its successors. In addition, IRLS is recommended by M.S. Goldberg and A. Tuow (Reference 8) as the technique of choice for deriving learning curves.

The fact that Wedderburn did not claim error-minimizing properties for IRLS leads one to ask whether or not IRLS does, in fact, produce the minimum-percentage-error CER among all possible unbiased CERs for a given set of cost data. As it turns out, IRLS does not yield percentage-unbiased CERs that have the smallest possible percentage standard error. For such CERs, we must look elsewhere.

B. MINIMUM-PERCENTAGE-ERROR, ZERO-PERCENTAGE-BIAS CERs

The minimum-percentage-error, zero-percentage-bias (MPE-ZPB) idea was first floated in a paper at the U.S. Army Conference on Applied Statistics in 1998 (Reference 3), as a solution to the open (at the time) question of whether or not IRLS CERs had minimum possible percentage error as well as zero percentage bias. Since then, it has been demonstrated in all examples tested that MPE-ZPB CERs (referred to by some organizations as “ZMPE” or “zimpy” CERs) have smaller standard error and the same zero percentage bias as the IRLS CER of the same algebraic form for the same set of data. MPE-ZPB CERs are also derived using numerical analysis techniques implementable in Excel® Solver and other software packages. It has been convincingly shown that, if we take the IRLS coefficients as our “initial” coefficient set, and then run Excel Solver under MPE-ZPB conditions, the solution is a new coefficient set that results in a smaller percentage standard error (Reference 6).

MPE-ZPB is a “constrained optimization” method, i.e., it seeks coefficients for which the resulting CER has smallest possible percentage error, ***subject to the constraint that*** its percentage bias be zero. In other words, the minimum-percentage-error CER is selected, not from among all possible CERs, but only from among those that are pre-selected in advance to have zero bias. The constrained optimization process can be easily carried out using Excel® Solver.

THE FOUR BASIC CER FORMS

Most CERs fall into one of four major algebraic categories: (1) the factor CER; (2) the linear CER; (3) the power CER; and (4) the triad CER. If only one cost driver is involved (the types of CERs we have been discussing so far), the algebraic expressions for these respective CER forms are displayed in Table 4 below.

Category	Cost	Cost Driver	Coefficients	Basic Algebraic Form
Factor	y	x	a	$y = ax$
Linear	y	x	a, b	$y = a+bx$
Power	y	x	a, b	$y = ax^b$
Triad	y	x	a, b, c	$y = a+bx^c$

Table 4. CER Categories and Algebraic Forms

In many situations, it is determined that a subsystem or component cost has more than one significant cost driver, e.g., weight, beginning-of-life power, and battery capacity for a power subsystem. In such cases, the resulting “multivariate” CER is typically a combination of the four basic CER forms listed in Table 1.

The “multiple linear” CER form is a combination of the univariate (i.e., one cost driver) factor and linear CERs. Its generic algebraic form is

$$y = a + bx + cw + dz + \dots,$$

where x, w, and z are cost-driver values and a, b, c, and d are coefficients.

The “multiple power” CER form is a combination of the univariate factor and power CERs. Its algebraic form is

$$y = ax^b w^c z^d \ldots,$$

where again x, w, and z are cost-driver values and a, b, c, and d are coefficients.

The "multiple triad" CER comes in two possible forms, either

$$y = a + bx^c + dw^e + fz^g + \ldots,$$

or

$$y = a + bx^c w^d z^e \ldots,$$

where again x, w, and z are cost-driver values and a, b, c, d, e, f, and g are coefficients.

It is no more difficult to write Excel® spreadsheets carry out the calculation for these multivariate CERs and their quality metrics than it is to write them for the original univariate CERs, so in the examples that follow, we will focus on univariate CERs.

EXAMPLE 1. OLS and IRLS/MPE-ZPB FACTOR CERs

By the term "factor CER," we mean a cost-estimating relationship (CER) of the form $y = ax$, where y is cost, x is the numerical value of a cost driver, and the coefficient a is the ratio between cost and cost driver. In rough-order-of-magnitude (ROM) CERs, the coefficient a typically represents dollars per pound, dollars per kilowatt, or something of that kind. More generally, in the case of so-called "cost to cost" CERs, a can represent dollars of system engineering, integration and testing, or program management ("SEIT/PM") per dollar of "prime mission product" (PMP), namely the developed or delivered hardware and software. In the latter case, y is the SEIT/PM cost and x is the PMP cost.

Univariate factor CERs have the unique characteristic that there are explicit algebraic formulas for the coefficient a in the case of OLS, IRLS, and MPE-ZPB, the latter two of which have the same solution. In the case of OLS, the formula for a is

$$a = \frac{\sum_{k=1}^{n} x_k y_k}{\sum_{k=1}^{n} x_k^2}.$$

In the cases of both IRLS and MPE-ZPB factor CERs, a is given by the formula

$$a = \frac{1}{n}\sum_{k=1}^{n}\frac{y_k}{x_k}.$$

This situation makes it particularly simple to derive univariate factor CERs from a set of cost and technical data, regardless of your personal philosophy on what the optimal CER characteristics are.

Further details, including mathematical derivations, about the factor-CER formulas are presented in Table 5.

Using the same data set as in the two earlier examples, we will now illustrate the factor CERs under both OLS and IRLS/MPE-ZPB criteria. The following set of computations implements the application of the OLS factor CER coefficient formula appearing above:

n	x	y	x^2	xy
7	7.9	5.900	62.410	46.610
	8.2	13.300	67.240	109.060
	9.8	43.595	96.040	427.231
	11.5	72.944	132.250	838.856
	16.4	95.434	268.960	1,565.118
	19.7	106.074	388.090	2,089.658
	23.6	116.274	556.960	2,744.066
Sums =	97.1	453.521	1,571.950	7,820.599

Table 5. Computations Leading to the OLS Factor CER's Coefficient *a*.

According to the formula, the coefficient of the OLS factor CER is ***a* = *7820.599/1571.950* = *4.975***. The quality metrics associated with the OLS factor CER appear in the table on the left side of Figure 9, and the graph of the CER is superimposed on the scattergram of the data point on the right side.

a =	4.975
R^2 =	87.1589%
Std Error =	20.016
Bias =	4.223
%Std Error =	46.5297%
%Bias =	15.8934%

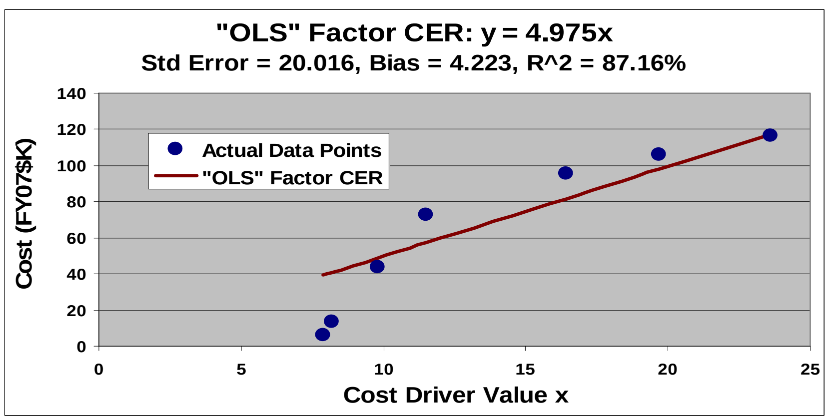


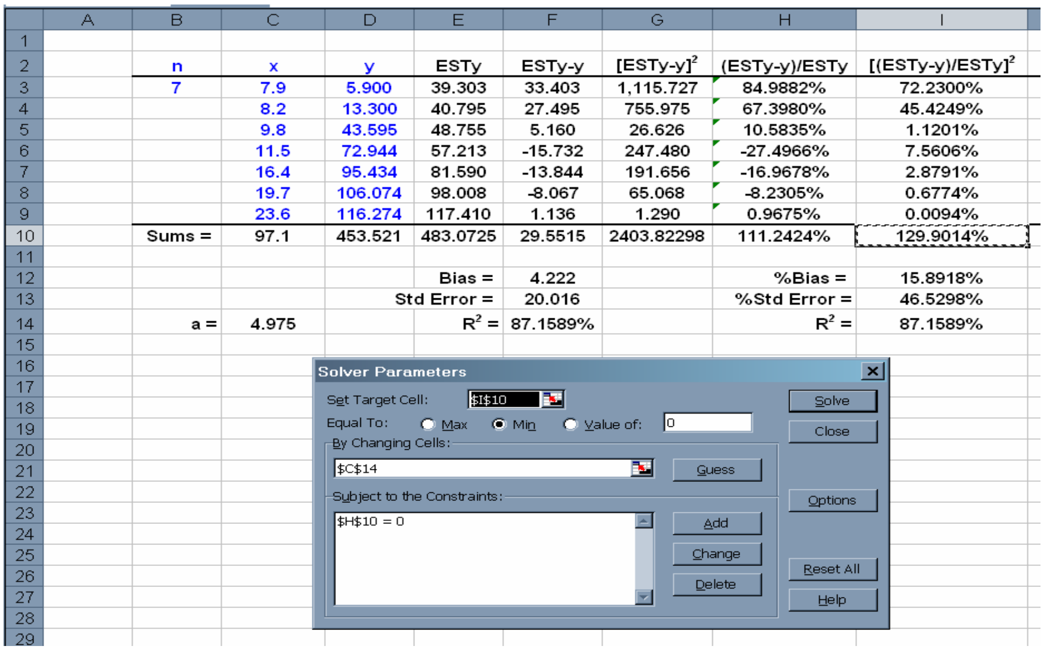
Figure 9: Quality Metrics and Graphical Representation of the OLS Factor CER.

As a sidelight to the OLS factor CER's quality metrics, we have appended the percentage standard error and bias information for comparison with those of the IRLS/MPE-ZPB factor CER, which we shall present next.

n	x	y	y/x
7	7.9	5.9	0.747
	8.2	13.3	1.622
	9.8	43.595	4.448
	11.5	72.944	6.343
	16.4	95.434	5.819
	19.7	106.074	5.384
	23.6	116.274	4.927
Sums =	97.1	453.521	29.291

Table 6. Computations Leading to the IRLS/MPE-ZPB Factor CER's Coefficient *a*.

According to the formula, the coefficient of the IRLS/MPE-ZPB factor CER is ***a = 29.291/7 = 4.184***. There is no other IRLS/MPE-ZPB CER form whose coefficients can be derived using an explicit algebraic formula. All other MPE-ZPB CERs, for example, must be derived using a mathematical optimization scheme, such as that carried out by Excel® Solver. Figure 10 below displays the inputs to the Solver dialogue box as it is being set up to solve the MPE-ZPB problem.



	A	B	C	D	E	F	G	H	I
1									
2		n	x	y	ESTy	ESTy-y	$[ESTy-y]^2$	(ESTy-y)/ESTy	$[(ESTy-y)/ESTy]^2$
3		7	7.9	5.900	39.303	33.403	1,115.727	84.9882%	72.2300%
4			8.2	13.300	40.795	27.495	755.975	67.3980%	45.4249%
5			9.8	43.595	48.755	5.160	26.626	10.5835%	1.1201%
6			11.5	72.944	57.213	-15.732	247.480	-27.4966%	7.5606%
7			16.4	95.434	81.590	-13.844	191.656	-16.9678%	2.8791%
8			19.7	106.074	98.008	-8.067	65.068	-8.2305%	0.6774%
9			23.6	116.274	117.410	1.136	1.290	0.9675%	0.0094%
10		Sums =	97.1	453.521	483.0725	29.5515	2403.82298	111.2424%	129.9014%
11									
12					Bias =	4.222		%Bias =	15.8918%
13					Std Error =	20.016		%Std Error =	46.5298%
14		a =	4.975		R^2 =	87.1589%		R^2 =	87.1589%

Figure 10: Excel® Solver Ready to Calculate MPE-ZPB Factor CER's Coefficient.

This is the same spreadsheet after Solver has been exercised (by clicking the "Solve" box) to obtain the coefficient value of 4.184:

	A	B	C	D	E	F	G	H	I
1									
2		n	x	y	ESTy	ESTy-y	$[ESTy\text{-}y]^2$	(ESTy-y)/ESTy	$[(ESTy\text{-}y)/ESTy]^2$
3		7	7.9	5.900	33.057	27.157	737.483	82.1518%	67.4893%
4			8.2	13.300	34.312	21.012	441.502	61.2380%	37.5009%
5			9.8	43.595	41.007	-2.588	6.698	-6.3112%	0.3983%
6			11.5	72.944	48.120	-24.824	616.210	-51.5864%	26.6115%
7			16.4	95.434	68.624	-26.810	718.781	-39.0682%	15.2632%
8			19.7	106.074	82.432	-23.642	558.927	-28.6800%	8.2254%
9			23.6	116.274	98.751	-17.523	307.039	-17.7441%	3.1485%
10		Sums =	97.1	453.521	406.3037	-47.2173	3386.64041	0.0000%	158.6372%
11									
12					Bias =	-6.745		%Bias =	0.0000%
13					Std Error =	23.758		%Std Error =	51.4194%
14		a =	4.184		R^2 =	87.1589%		R^2 =	87.1589%

Solver Results

Solver found a solution. All constraints and optimality conditions are satisfied.

Reports: Answer, Sensitivity, Limits

Keep Solver Solution

Restore Original Values

OK | Cancel | Save Scenario... | Help

Figure 11: Excel® Solver Exercised to Calculate the MPE-ZPB Factor CER's Coefficient.

Note the quality metrics associated with the MPE-ZPB factor CER. They are listed in the table on the left side of Figure 12, with the CER's graph appearing on the right side.

a =	4.184
R^2 =	87.1589%
%Std Error =	51.4194%
%Bias =	0.0000%

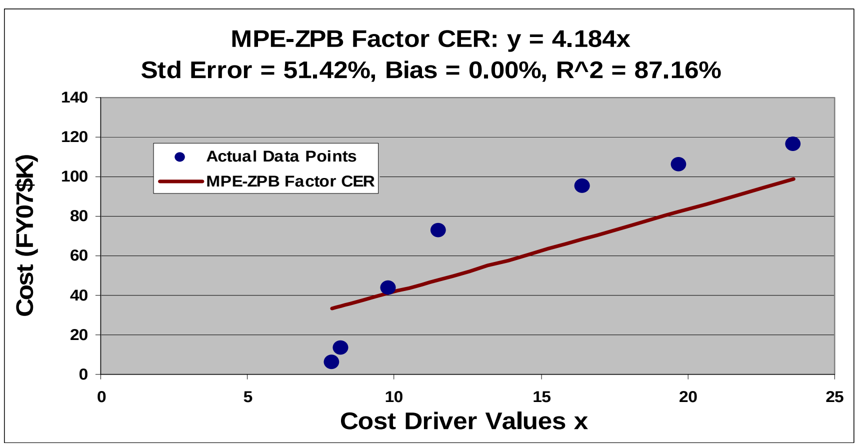


Figure 12: Quality Metrics and Graphical Representation of the MPE-ZPB Factor CER.

A comparison of the OLS and MPE-ZPB factor CER quality metrics and graphics is instructive. Notice from Figure 9 that the OLS CER's percentage standard error is smaller than that of the MPE-ZPB CER, although its bias is positive and rather large (15.89%). When the bias is reduced to zero, it can be seen in the graph of Figure 8 that the MPE-ZPB line is somewhat lower than the OLS line of Figure 9, noticeable especially in their respective positions relative to the largest actual data point.

EXAMPLE 2. IRLS and MPE-ZPB LINEAR CERs

A linear CER has both a nonzero (usually) constant term plus a constant multiple of the cost-driver value. That means a linear CER can represent either an extension of the situation that leads to a factor CER, e.g., a certain amount of dollars for each pound of the components weight plus a fixed cost not related to weight or anything else, or an entirely new situation in which the cost starts out at some nonzero fixed-cost value and rises linearly with some cost driver. The linear CER's algebraic form, as we have noted earlier, is $y = a+bx$, where y is cost, x is the numerical value of a cost driver, and the coefficients a and b represent the intercept and slope of the CER.

We already have discussed the OLS case, where there are explicit formulas for the coefficients a and b. In the case of IRLS and MPE-ZPB, however, there are no such formulas, so we shall illustrate the use of an Excel® spreadsheet that implements the required mathematics.

As it turns out, in the case of IRLS we can go partway toward finding explicit formulas for the coefficients a and b. The mathematical details can be found in the appendix, but in Figure 13 below, we illustrate the results of setting up and exercising an Excel® spreadsheet that carries out the required computations. We again use the same set of cost data that we have used before.

Iteration #	0	...	12
a =	-29.499	...	-85.094
b =	6.797	...	11.776
%Bias =	7.2519%	...	0.0000%
%Std Error =	47.6226%	...	38.3102%
R^2 =	87.1589%	...	87.1589%

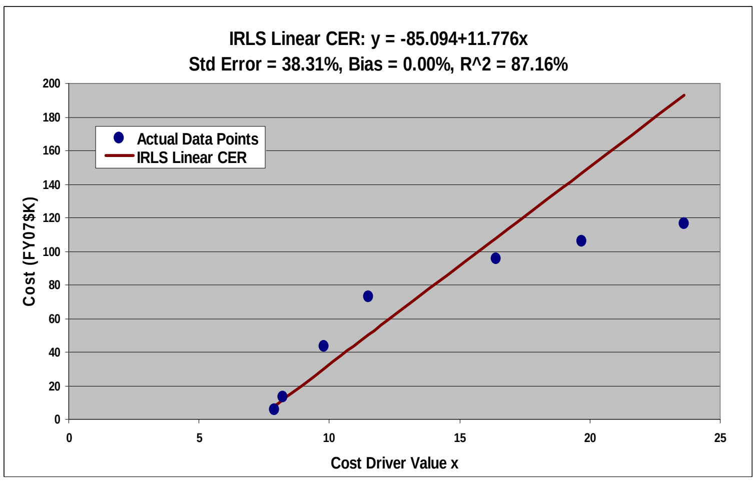


Figure 13: IRLS Linear CER and Quality Metrics.

The initial (starting) values of a and b (Iteration #0 on the left side of Figure 9) have been selected by using the parameters of the linear CER derived on the basis of OLS criteria several pages back. With IRLS we are working with percentage, rather than dollar-valued, error and forcing the percentage bias to zero. Comparing the results in Figure 13 with those in Figure 2, we see that the percentage-error criterion of Figure 13 moves the CER closer to the lower-cost data points and farther from the higher-cost data points. This is the effect of equalizing the percentage error across the data base. In Table 2, the dollar-valued error is equalize across the data base, leading to a CER that is closer to the higher-cost points and (relatively) farther from the lower-cost points, but in fact the same absolute distance from both.

For the MPE-ZPB linear CER, we use a spreadsheet format based on Excel® Solver much like that of Figures 10 and 11 that we applied to the case of the MPE-ZPB factor CER. Our starting point for the optimization will be the values of the ***a*** and ***b*** coefficients of the IRLS linear CER, because we want to verify that the MPE-ZPB process will indeed improve the CER's quality metrics.

a =	-91.041
b =	12.423
R^2 =	87.1589%
%Bias =	0.0000%
%Std Error =	38.0323%

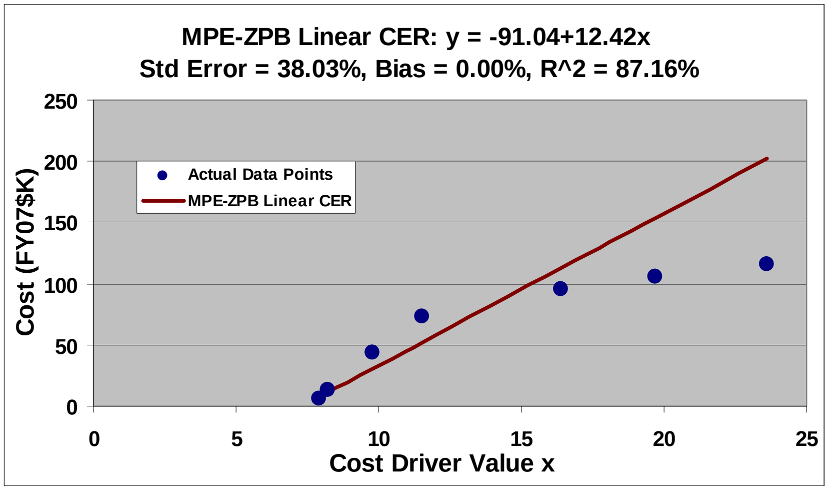


Figure 14: MPE-ZPB Linear CER and Quality Metrics.

Notice from the table of quality metrics in Figure 14 that the percentage standard error of 38.03% is somewhat lower in the case of the MPE-ZPB linear CER than it is in the case of the IRLS linear CER, which is 38.31%, as shown in Figure 10. Given that it takes less effort to derive an MPE-ZPB CER than it does to derive an IRLS CER, this is a useful fact to know.

Note also that all linear CERs, factor CERs included, have the same R^2 value. This is due to the fact that R^2 actually measures the linearity of the points in the data base with respect to the linearity of the CER. Since we are using the same data base, the relative linearity of the data base is the same with respect to any linear CER. This is about to change, though, as we will soon begin working with nonlinear CERs again. Recall that the R^2 value of the power CER describe in Table 3 and Figure 8 was, at 75.52%, considerably lower than the 87.16% that characterizes all the linear CERs.

Before we leave this subject, we should point out that Excel® Solver is only the most readily available computational aid that can be used to carry out IRLS- and MPE-ZPB-required calculations. In addition to more capable optimization routines, some of which are marketed as high-capability add-ons to Microsoft Excel®, several professional statistical computing packages that do IRLS calculations are commercially available. For analysts who do a lot of CER development, it might very well be worthwhile to investigate the tools available to professional statisticians. As noted earlier, however, it is not always required that numerical analysis software such as Excel® Solver be applied to derive IRLS CERs. Reference 5 shows how the coefficients of factor CERs in particular, i.e., those of the form $y = ax$, can be worked out explicitly.

EXAMPLE 3. MPE-ZPB POWER CER

Several pages back, we used the method of logarithmic transformation followed by OLS applied to the logarithms of ***x*** and ***y*** to find a power CER of the form $\boldsymbol{y = 0.123x^{2.304}}$ for our data set, with $\boldsymbol{R^2 = 75.5173\%}$, percentage standard error of ***71.7553%***, and bias of ***-14.7891%***. Now we apply the MPE-ZPB technique to obtain a percentage-unbiased CER that has smallest possible percentage standard error.

Notice that, in addition to the smaller percentage standard error (54.0456% vs. 71.7553%), zero percentage bias (vs. -14.7891%), and higher $\boldsymbol{R^2}$ (83.3430% vs.75.5173%), the MPE-ZPB CER is less convex than the log-log-derived CER and so "bends" closer to the points than the latter. The MPE-ZPB CER is still handicapped by having to pass through the origin, so it's convexity cannot more closely match the convexity pattern of the actual data points, but it does get as close to them as possible.

a =	1.286
b =	1.445
R^2 =	83.3430%
%Std Error =	54.0456%
%Bias =	0.0000%

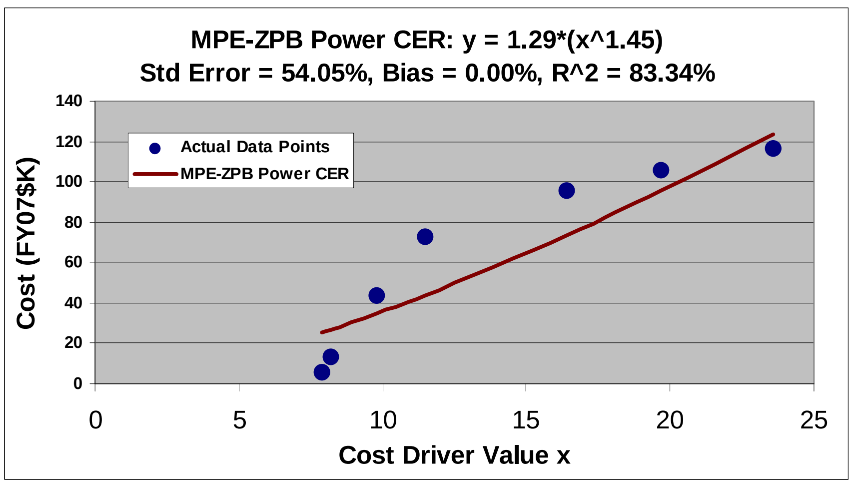


Figure 15: Solution to the Power MPE-ZPB CER Problem Using Excel® Solver

EXAMPLE 4. MPE-ZPB TRIAD CER

Finally, we arrive at the case of the "triad" CER $\boldsymbol{y = a + bx^c}$. One way to describe this kind of CER is that it is a power CER that is not required to pass through the origin and so can track data points of any convexity tendency equally well. Furthermore, the linear CER is a special case of the triad CER obtained by setting the exponent $\boldsymbol{c = 1}$. In other words, the triad CER becomes a power CER when $\boldsymbol{a = 0}$ and a linear CER when $\boldsymbol{c = 1}$. Allowing both ***a*** and ***c*** to assume numerical values different from 0 and 1, respectively, means that the triad CER offers the benefits of both the power (curvature) and linear (nonzero intercept) CER classes without imposing the disadvantages of either.

a =	-2,865.297
b =	2,609.387
c =	0.046
R^2 =	94.0345%
%Std Error =	26.1552%
%Bias =	0.0000%

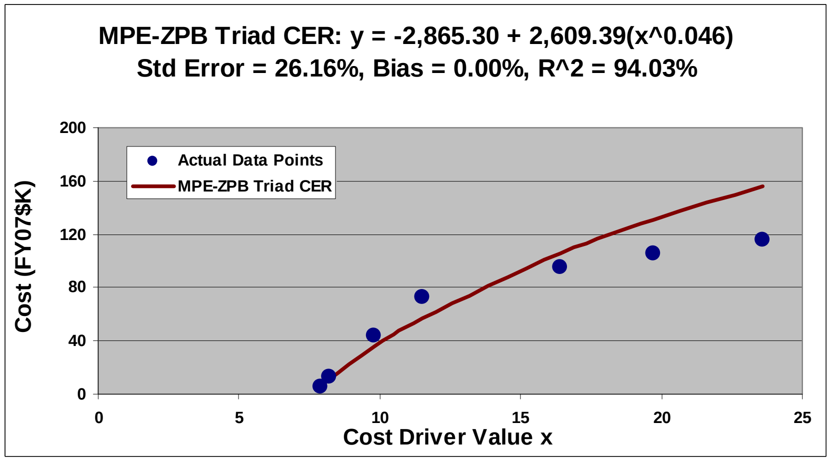


Figure 16: Linear MPE-ZPB CER Setup Prior to Activation of Solver Using Visual Choices of Initial Coefficient Values

Note that triad CER is superior to both the linear and power CERs, both visually with respect to tracking the data points and on the basis of the CER quality metrics. It has a higher R^2, a lower percentage standard error, and the same zero bias.

IV. CER SELECTION

When a CER has passed its evaluation, it is ready for application. A CER may be used as a primary estimating method to forecast costs, or to cross check an estimate developed using another estimating technique. For example, an analyst may have generated an estimate using a grassroots approach (e.g., a detailed build-up by hours and rates), and then used a CER estimate based on the same data as a sanity test of the grassroots' results. A CER can provide more realistic estimates than grass roots approaches if the latter are not closely and objectively tied to actual cost history.

A CER developed to make a specific forecast may be used with far more confidence than a "generic" CER developed for a wider range of applications. Care must be especially taken in using a generic CER when the characteristics of the forecasting universe are, or are likely to be, different from those of the CER database used to build it. A generic CER may have to be revalidated or modified for use in a particular application, and the changes made to it documented.

In order to apply good judgment in the use of CERs, the analyst needs to know their strengths and weaknesses.

Strengths

1. Mathematical derived CERs based on historical data can claim "objectivity."
2. CERs reveal sensitivity of cost to physical and performance attributes
3. Use of valid CERs can reduce proposal preparation, evaluation, negotiation costs, and cycle time, particularly with regard to low-cost items that are time and cost intensive to estimate using other techniques.

4. Given a CER equation and the required input parameters, developing an estimate is a quick and easy process.

5. Most CERs can be used with a small amount of top-level information about the product or service being estimated.; consequently, CERs are especially useful in the research, development, test and evaluation (RDT&E) phase of a program.

Weaknesses

1. CERs may be too simple to be used to estimate certain costs. When detailed information is available, a detailed estimate may be more reliable than one based on a CER.

2. Historical data are difficult to obtain and properly normalize. Problems with the database may mean that a particular CER should not be used. While the analyst developing a CER should also validate both the CER and the database, it is the responsibility of the estimator to determine whether it is appropriate to use a CER in given circumstances by reviewing its source documentation. The user should determine what the CER is supposed to estimate, what data were used to build it, how current are the data, and how the data were normalized. Never use a CER or cost model without reviewing the source documentation.

3. CERs cannot account for "New" Ways of Doing Business.

4. CERs cannot estimate costs of Beyond-State-of-the-Art technology.

Examples of CERs in Use

This section contains examples of CER provided by contractors who participated in the Parametric Estimating Reinvention Laboratory. A CER calculates changes in prices or costs (in constant dollars) as some physical, performance, or other cost-driving parameter changes. Such a relationship may be applied to a variety of items and services.

A. Construction

Many construction contractors use a rule of thumb that relates floor space to building cost. Once a general structural design is determined, the contractor or buyer can use this relationship to estimate total building price or cost, excluding the cost of land. For example, when building a brick two-story house with a basement, a builder may use $60/square foot to estimate the price of the house. Assume the plans call for a 2,200 square foot home. The estimated build price, excluding the price of the lot, would be $60/sq. ft. x 2,200 sq. ft. = $132,000.

B. Electronics

Manufacturers of certain electronic items have discovered that the cost of a completed item varies directly with the number of total electronic parts in it. Thus, the sum of the number of integrated circuits in a specific circuit design may serve as an independent variable (cost driver) in a CER to predict the cost of the completed item. Assume a CER analysis indicates that $57.00 is required for set-up, and an additional cost of $1.10 per integrated circuit required. If evaluation of the engineering drawing revealed that an item was designed to contain 30 integrated circuits, substituting the 30 parts into the CER gives:

Estimated item cost = $57.00 + $1.10 per integrated circuit * number of integrated circuits

= $57.00 + $1.10 (30)

= \$57.00 + \$33.00
= \$90.00

C. Weapons Procurement

CERs are often used to estimate the cost of the various parts of an aircraft, such as that of a wing of a supersonic fighter. Based on historical data, an analyst may develop a CER relating wing surface area to cost, finding that there is an estimated \$40,000 of wing cost (for instance, nonrecurring engineering) not related to surface area, and another \$1,000/square foot that is related to the surface area of one wing. For a wing with 200 square feet of surface area:

Estimated price = \$40,000 + (200 sq ft x \$1,000 per sq. ft.)
= \$40,000 + 200,000
= \$240,000

Examples of Simple CERs

Figure 17 provides examples of simple CERs implemented by various companies in the Parametric Estimating Initiative Reinvention Laboratories.

CER Title	Pool Description	Base Description	Application
Panstock Material	Allocated panstock dollars charged.	Manufacturing assembly "touch" direct labor hours charged.	Panstock is piece-part materials consumed in the manufacturing assembly organization. The panstock CER is applied to 100% of estimated direct labor hours for manufacturing assembly effort.
F/A-18 Software Design Support	Allocated effort required performing software tool development and support for computer & software engineering.	Computer and software engineering direct labor hours charged.	F/A-18 computer and software engineering support direct labor hours estimated for tool development.
Design Hours	Design engineering including analysis and drafting direct labor hours charged.	Number of design drawings associated with the pool direct labor hours.	The design hours per drawing CER is applied to the engineering tree (an estimate of the drawings required for the proposed work).
Systems Engineering	Systems engineering (including requirements analysis and specification development), direct labor hours charged.	Design engineering direct labor hours charged.	The system engineering CER is applied to the estimated design engineering direct labor hours.
Tooling Material	Nonrecurring, in-house, tooling raw material dollar costs charged.	Tooling nonrecurring direct labor hours charged.	The tooling material CER is applied to the estimated nonrecurring tooling direct labor hours.
Test/Equipment Material (dollars for avionics)	Material dollars (<\$10k)	Total avionics engineering procurement support group direct labor hours charged.	The test/equipment material dollars CER is applied to the estimated avionics engineering procurement support group direct labor hours.

Figure17: Examples of Simple CER

Summary of CER Evaluation

The following list suggests additional questions which might be asked about a CER in order to determine its limitations and applicability. Consider the importance of the costs which a CER estimates when using the questions -- don't spend a lot of time asking them, or getting their answers, for example, when the CER's result is a minor cost, or is lost in rounding when rolled into higher-level estimate.

1. What proportion of the estimate is directly affected by the CER?
2. How much precision is needed for the total estimate, and for the part of it affected by the CER?
3. Is there a logical relationship between a CER's dependent variable and its independent variables?
4. Is this relationship functional or statistical? If functional, what is it, and why? If statistical, does the associated data support the CER's intended application?
5. Are relationship and the independent variables statistically significant? At what level of confidence?
6. What happens to the estimate when reasonable variations of the input parameters are used?
7. Are the analytical methods and techniques used to develop and use the CER sound and appropriate?
8. Does the CER generate the type of estimate required?
9. Are the model input parameters available and reliable in the phases of the system life cycle when it will be used?
10. Are the concepts behind the CER widely accepted in industry and generally understood?
11. Are the CER's strengths and limitations reasonable?
12. What is the effect of input uncertainty on the estimate's confidence interval?
13. Are the mathematical procedures used to develop the CER rigorous?
14. Does the CER integrate information from other systems?
15. Is the CER compatible with other CERs/models in theory and operation?
16. Is a sufficient amount of accurate and relevant historical data available for model development?

17. Are the cost estimates made with the model consistent with user/contractor performance?
18. Does the CER model documentation provide insight into historical data?
19. What parametric development concepts does the CER incorporate?
20. Are the developing organization's estimating systems and policies current?
21. Are the CER's source data verifiable?
22. Does the developing organization have written guidelines for the development and support of parametric estimates?
23. How are users trained to use the CER?
24. How is the CER updated?
25. Do the CER's parameters adequately describe the item/service which is estimated?
26. Are the engineering input decisions that contributed to the CER development documented?
27. How difficult is it to use the CER?
28. Is the CER flexible (e.g., to changing programmatic and technical issues, or parameters)?
29. Is the CER model useful at varying levels of input detail?
30. Can the CER be used across a range of time, products, and technology changes?
31. How easy is it to misuse the CER?
32. Does the CER avoid personal or organizational bias?
33. Can the CER results be adjusted?
34. Does use of the CER require experienced analysts and/or special training?
35. Have the CER's results been checked against test cases?
36. Are the CER's results in the correct format and level of detail?

V. Validation

A. Validation Requirements

A CER, as any other parametric estimating tool, must produce, to a given level of confidence, results within an acceptable range of accuracy. It must also demonstrate estimating reliability over a range of data points or test cases. The validation process ensures that a CER meets these requirements. Since a CER developer and customer must, at some point, agree on the validation criteria for a new CER, the Parametric Estimating Reinvention Laboratory determined that the use of an IPT is a best practice for reviewing and implementing it. The contractor, buying activity, DCMA, and DCAA should be part of the IPT. Chapter _, provides detailed guidance.

Figure 18 illustrates the validation process flow, which incorporates the CER testing methodology discussed earlier in the chapter. The process, described in Figure 19, is a formal procedure which a company should use when developing and implementing a CER. It describes the activities and criteria for validating simple CERs, complex CERs, and parametric models. Figure 20 contains the guidelines for statistical validation.

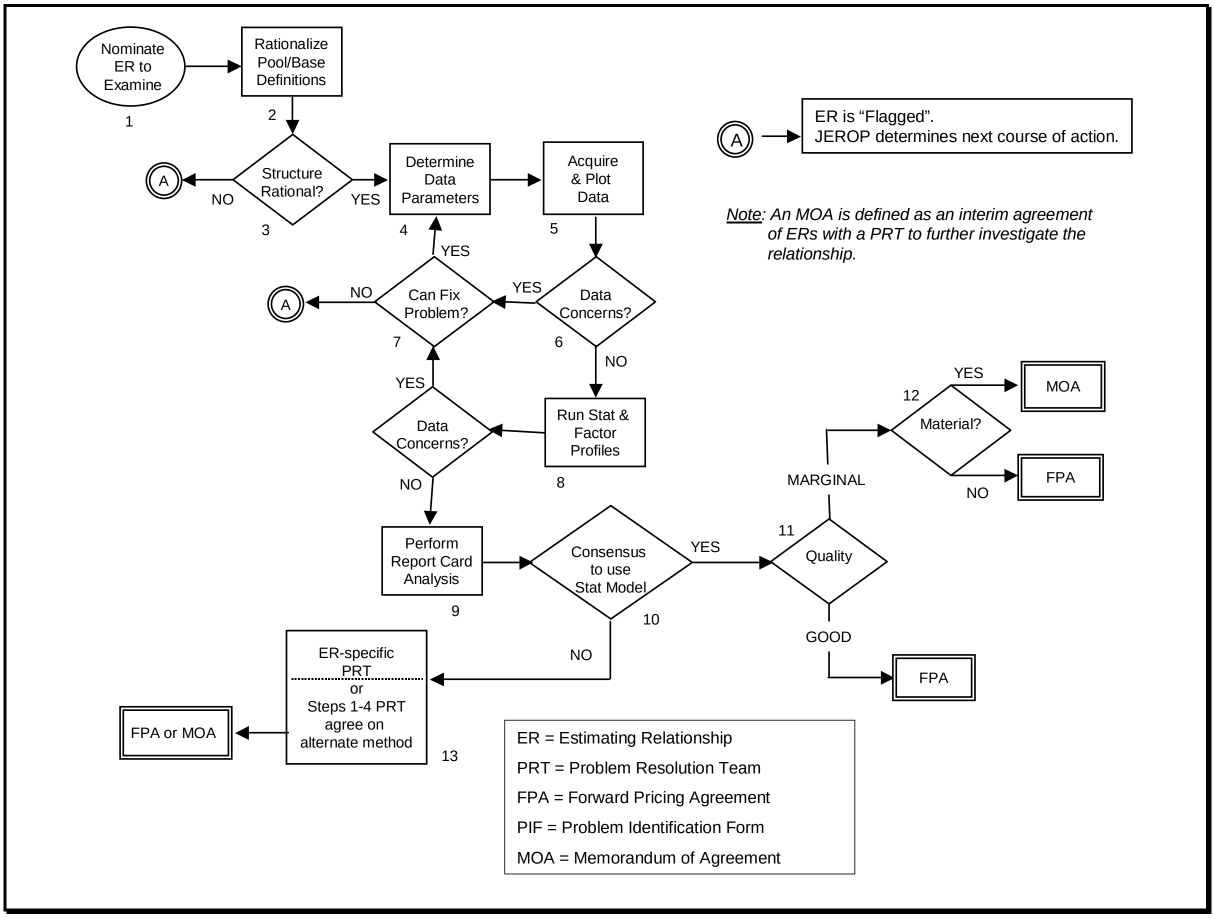


Figure 18: Estimating Relationship Validation Process

Discussion of Activities

1 - Assess materiality.
2 - Examine rationale and data, or use additional historical data.
3 - Investigate alternative forms.
4 - Team is encouraged to review data beyond that used to develop the current CER, i.e., additional completed jobs for Steps 1 & 2 CERs, or longer time periods for Steps 3 & 4 CERs.
  - Multivariate (more than one independent variable) solutions may be considered.
  - Examine ER across programs for rationalization of differences.
5 - Team may explore linear, logarithmic, exponential, polynomial, power, moving average, or any other model structures implied by the data patterns and/or rationale of the effort.
6 - Check for evidence of outliers, influentials, time trends, bi-modal data, etc.
7 - Team analyzes data sources.
8 - Develop results based on weighted factor methodology and linear regression with intercept, unless otherwise agreed.
  - Construct 'report card' with F-stat, R-squared, CV, and narrative for stat method; with MAD and narrative for factor method.
  - Plot results. Analyze residuals, checking for patterns in the residuals to ensure that the regression assumptions were not violated. Examine raw versus fitted data for outliers, using a rule of thumb of 2 to 3 standard deviations as a means of flagging data points for further investigation.
9 - Team analyzes 'report card' for ER based upon guidance shown in Figure 3-9.
10 - Team decides by consensus whether one or more of the methods presented are acceptable. Unless a compelling argument is presented by one of the organizations, the statistical model is to be preferred. Lack of consensus among the three organizations, or consensus that no available model is satisfactory, results in process flow to Step 13.
11 - Qualitative decision by team determining whether stat model is "Good" or "Marginal", using report card criteria as guide.
12 - Team determines materiality of the ER based on dollar impact, breadth of application, etc.
13 - Alternative methods include, but are not limited to, other statistical models, simple or weighted averaging and other factors, discreet estimating, accounting changes, investigation of other options for 'base', etc.

(A) Flagging:

A. PIF should be completed and forwarded to JEROP.
B. JEROP will determine next action, including possibility of establishing PRT to study ER.
C. If tasked with remedial investigation, PRT may:
   1. Assess materiality.
   2. Re-examine rationale and data, or use additional historical data.
   3. Investigate alternative forms.
D. PRT will determine requirements for accepting, rejecting, or restructuring ER.

Figure 19: Estimating Relationship Validation Process Activities

Summary of ER Report Card Criteria

This 'report card' is a summary of the key attributes of the statistically derived model and of the weighted factor, and serves as a starting point for the qualitative analysis of the proposed estimating tool.

The *p*-values of the F-test and of the t-test are the most critical, being viewed as essentially pass/fail. The other criteria, including the comments in the narrative portion of the report card, should be weighed in composite to determine the acceptability of the tool. This overall qualitative opinion should weigh the quality of the statistical results against the materiality of the effort and the quality of possible alternative methods.

		Good	Marginal
Statistically Derived ER:	*p*-value of the F-test:	≤ 0.10	≤ 0.15
	p-value of the t-test:	≤ 0.10	≤ 0.15
	Coefficient of Variation (CV) :	≤ 0.25	0.25 → 0.30
	R-squared:	≥ 0.70	0.35 → 0.70
	Narrative: *This section of the report card should be used to record other pertinent information, particularly non-quantitative information, about the effort to be modeled or about the proposed estimating tool. For example, data constraints, materiality, exogenous influences, etc., may impact the acceptability of the proposed tool.*		
Weighted Factor:	MAD as % of ER mean:	≤ 0.25	0.25 → 0.30
	Narrative: *- same as above for statistically derived model -*		

Terminology:

- F-test: Tests for trend in the data versus random dispersion.
- t-test: Measures the significance of the individual components of the model; where there is only one independent variable (one 'base' variable), the significances of the t-test and of the F-test are identical.
- R-squared: Measures the percentage of variation in the pool explained by the CER or model; varies between 0% and 100%.
- CV: Coefficient of variation is a measure of dispersion; produces a measure of 'average estimating error'.
- MAD: Mean absolute deviation is a measure of dispersion comparing how well the individual point relationships match the mean relationship of the composite data.

Figure 20: Summary of Estimating Relationship Report Card

VI. Common issues and pitfalls in CER Development

REFERENCES

1. Book, S.A. and P.H. Young, "Optimality Considerations Related to the USCM-6 'Ping Factor'," ICA/NES National Conference, Los Angeles, CA, 20-22 June 1990, 40 briefing charts.

2. Book, S.A. and P.H. Young, "General-Error Regression for Deriving Cost-Estimating Relationships," *The Journal of Cost Analysis*, Fall 1997, pages 1-28.

3. Book, Stephen A. and Lao, Norman Y., "Minimum-Percentage-Error Regression under Zero-Bias Constraints", *Proceedings of the Fourth Annual U.S. Army Conference on Applied Statistics, 21-23 October 1998*, U.S. Army Research Laboratory, Report No. ARL-SR-84, November 1999, pages 47-56.
4. Book, S.A. and P.H. Young, "The Trouble with R^2," *The Journal of Parametrics,* Summer 2006, pages 87-112.Book, S.A., "The Mathematics of Deriving Factor CERs from Cost

Data," *Parametric World*, Vol. 25, No. 3 (Summer 2006), pages 16-22.Book, S.A., "Unbiased Percentage-Error CERs with Smaller Standard Errors," *Journal of Cost Analysis & Management*, Winter 2006, pages 55-72.Eskew, H.L. and K.S. Lawler, "Correct and Incorrect Error Specifications in Statistical Cost Models," *Journal of Cost Analysis*, Spring 1994, pages 105-123.

9. Goldberg, Matthew S. and Touw, Anduin E., *Statistical Methods for Learning Curves and Cost Analysis*, Institute for Operations Research and the Management Sciences (INFORMS), 2003, 196 pages.

10. Hu, S.P. and A.R. Sjovold, "Error Corrections for Unbiased Log-Linear Least Square Estimates," Tecolote Research, Inc., Report No. TR-006/1, October 1987.

11. Nelder, J. A., "Weighted Regression, Quantal Response Data, and Inverse Polynomials," *Biometrics*, Vol. 24 (1968), pages 979-985.

12. Neyman, J. and E.L. Scott, "Correction for Bias Introduced by a Transformation of Variables," *Annals of Mathematical Statistics*, Vol. 31 (September 1960), pages 643-655.

13. Nguyen, P., N. Lozzi, W. Galang, et al., *Unmanned Space Vehicle Cost Model, Seventh Edition*, U.S. Air Force Space and Missile Systems Center (SMC/FMC), 2430 E. El Segundo Blvd., Suite 2010, Los Angeles AFB, CA 90245-4687, August 1994, xvi + 452 pages.

14. Young, P.H., "The Meaning of the Standard Error of the Estimate in Logarithmic Space," *The Journal of Cost Analysis & Management*, Winter 1999, pages 59-65.. Wedderburn, R.W.M., "Quasi-likelihood Functions, Generalized Linear Models, and the Gauss-Newton Method," *Biometrika*, Vol. 61, Number 3 (1974), pages 439-447.

APPENDIX C
Frequently Asked Questions

This appendix discusses the most frequently asked questions (FAQs) of Reinvention Laboratory participants and observers concerning the implementation, evaluation, and negotiation of parametric estimates. The PCEI Working Group, which consisted of representatives from Industry and Government, developed answers to the FAQs. Since the Working Group is not a policy-making organization, some answers require additional guidance from Government policy or decision makers.

Regulatory Issues

How does the Truth in Negotiations Act (TINA) apply to parametric cost estimating techniques?

TINA requires that cost or pricing data be certified as current, accurate, and complete as of the date of negotiation, or an agreed-to date as close as practical to the date of negotiation. This applies to parametric estimating techniques if they are used to develop a proposal that requires cost or pricing data (i.e., proposal exceeding $550,000), unless one of the exceptions at FAR 15.403-1(b) applies.

A contractor with adequate parametric estimating system policies and procedures that are appropriately followed should comply with TINA. Like all estimating techniques, it is important to develop parametric-based estimates using data that is current, accurate, and complete as of the date of negotiation or another agreed-to date. Estimating procedures should ensure that a contractor updates its databases periodically based on a disciplined approach (e.g., yearly, quarterly, monthly) to maintain currency. The definition of what is current will depend on the nature of the data. Annual updates may be adequate for some kinds of data while monthly or more frequent updates may be required for others. It is important to have procedures for identifying significant data outside the normal cycle for analysis, to determine if out-of-period updates are needed to maintain currency of the estimate. Chapter 7, Government Compliance, discusses TINA issues in more detail.

Are estimates developed using parametric techniques subject to certification as cost or pricing data?

A decision concerning the need for a certification requirement depends on the nature of the procurement and if standards in FAR 15.403 apply. Certification is not required if the procurement is based on adequate price competition, prices are set by law or regulation, or the product is a commercial item as defined in FAR 2.101. Otherwise, certification is required unless a waiver is granted as stated in FAR 15.403(c)(4). Cost and pricing data consist of all the facts that can reasonably contribute to the soundness of future cost estimates and to the validity of costs already incurred. For parametrics, factual data includes the historical data used in calibrating the model (e.g., technical, programmatic, cost), and information on management decisions that may significantly affect costs (e.g., significant changes to management and manufacturing processes). Chapter 7, Government Compliance, provides additional information on this subject.

Can parametric techniques be used to perform price or cost analysis?

FAR 15.404-1(b)(2)(iii) states that parametrics are an acceptable method for performing price analysis. Similarly, FAR 15.404-1(c)(2)(i)(C) indicates that a variety of techniques can be used to perform cost analysis. Therefore, parametric techniques can be used by the Government to determine a reasonable price using contractor history, Government history, other knowledge bases, and cost models that have been calibrated and validated. The outputs from these models can be used as independent cost estimates from which the validity of offered prices can be compared.

Similarly, a prime contractor can use an appropriately calibrated and validated parametric model to evaluate a subcontractor's submission. If a prime contractor has a reliable model with which to analyze subcontractor data, then this tool can be used to satisfy the prime contractor's responsibility to perform price or cost analysis. In certain situations, a prime contractor will be required to disclose the subcontractor's cost or pricing data (including calibration and validation data) in its proposal, along with the results of its price or cost analyses. Chapter 8, Other Parametric Applications, includes an example related to the cost/price analysis of subcontracts.

Calibration and Validation

What is the general meaning of calibration and validation?

Calibration is the process of adjusting a commercial model's parametric values to reflect an organization's cost and product history. The calibration process converts a commercially developed (or public domain) model into the equivalent of one developed locally (i.e., by the organization). Validation is the process or act of demonstrating a model's ability to function as a credible forward estimating tool or system. Validation is performed for all parametric estimating techniques (e.g., CERs, models).

A parametric model should be viewed as part of an estimating system (discussed in Chapter 7, Government Compliance). The focus of all calibration and validation activities should be on the development of an adequate estimating system that reliably predicts future cost expenditures. A parametric estimating system includes: data upon which the technique is based; guidelines and controls established to ensure consistent and predictable system operation; procedures to enforce the consistency of system usage between calibration and forward estimating processes. To pass the validation test, the system must demonstrate its ability to provide credible forward estimates.

What elements should be included in the review of the cost drivers and supporting databases used in parametric models?

A review ensures that the contractor:

- Identifies the key input drivers and explains their relative impact on the estimate;
- Provides and supports the data and inputs used in calibration, and demonstrates that the calibrated model projects the contractor's historical costs within a reasonable degree of accuracy;
- Explains any adjustments to the model operation or to the key inputs, and provides adequate rationale for such adjustments;
- Demonstrates that the calibrated model produces reliable estimates in comparison to some other benchmark (e.g., actuals, comparative estimates).

How often should parametric technique databases be updated?

The frequency of database updates varies according to the parametric estimating techniques utilized and a contractor's product cycle. Assuming that a contractor's processes are stable, databases should be reviewed and updated at least annually.

Some considerations that might trigger more frequent updates are:

- New processes are being implemented that are expected to significantly impact cost;
- Cost estimating relationship values are fluctuating significantly between updates;
- An additional contract lot is completed, and there is a significant change in the unit cost.

It is important for contractors to have effective procedures to identify and analyze significant "out of cycle" data to determine if a special update is needed. Chapter 7, Government Compliance, discusses this in further detail.

Should a commercial model be used to prepare proposal submissions prior to being calibrated?

No. Parametric estimates based on commercial models should be calibrated before being used in any proposal. Uncalibrated commercial models are based on industry average data, which may not reflect a contractor's actual data and would most likely not produce accurate estimates. A commercial model should be calibrated so it that represents a given company's cost and culture; in addition, the contractor should use the same model version that is calibrated and validated to generate its estimates. Chapter 5, Complex Hardware Models, and Chapter 6, Complex Software Models, provide detailed guidance on calibration.

How does a contractor demonstrate that the use of parametric techniques result in accurate estimates?

This is the purpose of the validation process. A contractor should be able to demonstrate that its parametric technique(s) can reliably predict future cost expenditures when compared to some other benchmark (e.g., actuals, comparative estimates). Validation should also demonstrate that:

- Key personnel are experienced and have been adequately trained;
- Estimating procedures are available to ensure consistency of calibration and validation processes as well as proposal application;
- Cost model calibration has been performed and documented.

Validation is an on-going process. Contractors should establish a validation process to monitor the reliability of estimating techniques and models. Chapter 7, Government Compliance, provides further information on validation criteria.

What are the acceptable levels of accuracy?

Accuracy is a matter of judgment and should be based on a variety of factors. These include the significance of the cost being estimated, accuracy of alternate estimating techniques, contract type, and the level of risk a customer deems reasonable. Higher degrees of accuracy are generally expected for:

- Proposals for fixed price as compared to cost-type contracts;
- Follow-on production proposals, where cost history is known;
- Costs that are significant.

When accuracy is not within an acceptable level, the parametric technique should be reviewed. A contractor should have a monitoring system, which can identify and correct any problems of estimating accuracy. It is important to realize that many traditional estimating techniques (such as bottoms-up) do not provide validation data (i.e., accuracy of the estimates). Therefore, when evaluating the accuracy of parametric estimates, it is important to also consider the accuracy of the alternative estimating techniques.

How are savings associated with various cost-saving initiatives incorporated into estimates based on parametric techniques?

A contractor's estimating system policies and procedures should contain guidance for adjusting estimates to account for savings associated with new ways of doing business (e.g., effects of reorganizations, single process initiatives, post improvement initiatives, and software capability maturity models). Contractors incorporate such savings into their estimates in a variety of ways, including:

- Adjusting parameter values to reflect efficiencies;
- Deducting estimated savings from the bottom-line estimate;
- Updating (normalizing) databases.

All cost saving estimates should be logical, defensible, and reasonable. The estimates should also be adequately documented, and include the assumptions and rationale used to derive estimated savings. Chapter 7, Government Compliance, discusses the topic.

In the use of multi-variate regression, are there key statistical tests that are important to the acceptance of a CER?

When statistical techniques such as multi-variate regression are used to evaluate the strength of CERs, a variety of tests should be performed. Statistical tests that are frequently used include the F-test, t-test, Coefficient of Variation (CV), and the Coefficient of Determination (R^2). Chapter 3, Cost Estimating Relationships, and Appendix B, Detailed Math of Cost Estimating Relationships, provide additional information on statistical testing, and Appendix D, Related Web Sites and Supplementary Information, identifies sources of information about quantitative applications, including statistical analysis.

The results of a statistical analysis should not be the only criteria for determining whether a CER is acceptable. CERs (especially those deemed significant) should also be:

- Reliable and credible predictors of costs;
- Based on logical and strong data relationships;
- Derived from sound databases;
- Used according to estimating system policies and procedures;
- Monitored to ensure they remain good predictors.

What types of non-financial data are needed to calibrate a commercial model? How is this data normally gathered?

Non-financial data generally consist of technical information that describes the physical, performance, or engineering characteristics of a system, subsystem, or individual item. Examples include:

- Weights, electronic density, quantity, schedule, and complexity for hardware models;
- Size, reliability, language, and development processes for software models.

Technical data can be obtained from a variety of sources, such as enterprise resource planning (ERP) systems and/or management information systems; engineering drawings; and specifications. Technical data can also be obtained through interviews with technical personnel involved in the analysis, design, and manufacturing of existing and planned hardware and software programs. Chapter 2, Data Collection and Analysis, provides additional information.

Working Relationships

What is the suggested composition of IPTs which are implementing new parametric techniques?

IPTs should include representatives from a contractor's organization, the contractor's major buying activities, DCAA, and DCMA. The contractor is responsible for identifying and demonstrating the new parametric estimating opportunities, including how these will improve the estimating process (e.g., cost-benefit analysis). The contractor should also update estimating policies and procedures to incorporate new parametric tools and techniques (see Chapter 7, Government Compliance). The buying activity, DCAA, and DCMA team members provide real-time feedback to the contractor on such issues as estimating policies and procedures, calibration and validation criteria, and Government evaluation criteria. Chapter 8, Other Parametric Applications, has additional guidance on IPTs.

The Reinvention Laboratory recommends, as a best practice, that IPTs receive training in a newly implemented parametric technique. This includes general training (e.g., parametric estimating processes and team-building) as well as detailed instruction (e.g., focusing on the "how-tos"). Appendix D, Related Web Sites and Supplementary Information, contains training sources.

When should a company integrate its key customers with the IPT?

The contractor is responsible for integrating its major customer(s) with the IPT. When this is done depends on the contractor, how well defined its parametric estimating processes are, and whether the program to be estimated has been identified. The Reinvention Laboratory demonstrated, as a best practice, that key customer representatives should be included in the IPT before any proposal using a new parametric technique is submitted. Early customer involvement in the IPT, along with continuing communication among team members, is necessary for the successful implementation and acceptance of a parametric tool. Chapter 8, Other Parametric Applications, provides additional information.

If a contractor presents a parametric proposal to a buying activity that is not prepared to evaluate it, can the buying activity refuse the proposal?

A buying activity should not refuse a proposal because it is based on parametric techniques. If the activity does not have the expertise to evaluate such a proposal, it can obtain assistance from other sources, such as DCMA and DCAA. However, a buying activity may return a parametrically prepared proposal if a contractor does not provide timely supporting data during the proposal process or during fact finding (e.g., calibration and validation data, descriptions of the model (including key cost drivers), pertinent parts of disclosed estimating policies and procedures). Also, as discussed in the previous FAQ, the contractor should work with buying activity representatives in the IPT before submitting a proposal based on a new estimating technique. Chapter 8, Other Parametric Applications, considers this in more detail.

Who is responsible for deciding whether a parametric estimate should be accepted?

FAR 15.404-1(a)(1) states that contracting officers are responsible for evaluating the reasonableness of the offered prices. Their evaluations require the careful analysis of the proposals; development of negotiation positions; and, in sole source situations, negotiation of reasonable prices. Accepting a parametric estimate does not necessarily mean accepting the price offered. Each member of the IPT has an integral role in determining the acceptability of a specific technique and is, therefore, key to deciding whether a parametric estimate is acceptable. Chapter 8, Other Parametric Applications, discusses best practices for IPTs involved with parametric activities.

What should the buying activities expect from Industry when parametric techniques are used to develop estimates for proposals?

The buying activities should expect that the parametric techniques used to develop estimates for proposals have been adequately calibrated and validated by the contractor. The buying activities should also expect that the techniques are being used in the proper circumstances and are being applied consistently. In a parametrically prepared proposal, a contractor should provide adequate support, such as calibration and validation data, and descriptions of the key cost drivers. Chapter 7, Government Compliance, discusses proposal documentation requirements. Again, if customers are not familiar with a parametric technique, then the contractor should explain it to them well before using it in a proposal.

Miscellaneous

What impact should the use of parametric estimating techniques have on both Government and contractor resources?

The use of parametric estimating techniques should provide many benefits. The primary one is that – once the CER or model has been calibrated, validated, and implemented – both the contractor and the Government should save significant resources as compared with traditional bottoms-up estimating. For example, there should be substantial savings in personnel once the model is understood and accepted as a reliable predictor of costs, because less time will be needed to develop, review, and negotiate the resulting output. The Reinvention Laboratory found that another major benefit is the streamlining of the proposal update process. Other benefits include a reduction in cycle time for both Government and contractor representatives, and a reduction in costs related to proposal preparation, evaluation, and negotiation. Parametric estimating, when implemented effectively, is a valuable tool for doing more with less.

What is the biggest challenge related to evaluating and negotiating proposals based on parametric techniques, and how can these challenges be resolved?

Cultural resistance is the biggest challenge to the successful implementation of parametric techniques. Cultural resistance generally results from a lack of knowledge and understanding of parametric estimating; however, it can be overcome by educating the acquisition community on methods for implementing, evaluating, and negotiating proposals based on parametric techniques. This education can be provided in a number of ways. First, the results of the Reinvention Laboratory demonstrated that the use of an IPT process is a best practice for implementing new techniques because it provides team members with a good understanding of parametrics. Second, this Handbook is a useful reference for those involved with parametric activities since it contains chapters on the evaluation and negotiation of proposals, as well as examples from the Reinvention Laboratory. Third, a variety of parametric training courses are available (see Appendix D, Related Web Sites and Supplementary Information).

After receiving initial training on a particular parametric technique (e.g., CERs, company-developed model, or commercial model), how much experience does an evaluator need to become a proficient user?

While evaluators need to have a good understanding of the parametric techniques used to develop specific estimates, they do not need to be proficient users of the specific CER or model. Evaluators need to have a good understanding of a contractor's parametric estimating methodology. A company's estimating practices should contain guidance on topics such as key inputs, parameter values, and calibration and validation procedures. As discussed in the previous FAQ, use of an IPT process, this Handbook and training should help evaluators obtain a good understanding of the key issues related to parametrics. When outside expertise is needed, the major buying activities, DCMA and DCAA, have people experienced with parametric techniques who can provide additional support. For example, DCAA can provide financial advisory services to contracting officers on such issues as the adequacy of proposal submissions based on parametric techniques, and the validity of a contractor's parametric estimating system.

How can the buying activities communicate to Industry that the use of parametric techniques as a basis of estimate is acceptable?

The Reinvention Laboratory demonstrated that a contractor would be more likely to submit estimates based on parametric techniques if Requests for Proposals (RFPs) contained language permitting the use of parametrics. The following is an example of a RFP clause containing references to parametrics:

> "When responding to the cost volume requirements in the RFP, the offeror and its associated subcontractors may submit cost estimates utilizing appropriately calibrated and validated parametric models that are part of their disclosed cost estimating systems. These include contemporary estimating methods, such as cost-to-cost and cost to noncost estimating relationships (CERs), commercially available parametric cost models, and company-developed parametric cost models. If necessary, reasonable and supportable allocation techniques may be used to spread hours and/or cost to lower levels of the work breakdown structure (WBS). The offeror's use or non-use of the parametric estimating techniques for this proposal will not be a factor (positive or negative) in the evaluation of the offeror's response to the RFP. Cost estimates submitted using such parametric models should produce cost estimates that are reasonable and consistent and, as such, create a basis for negotiation of a fair and reasonable price."

APPENDIX D

Related Web Sites and Supplementary Information

This appendix contains Internet addresses for professional societies associated with cost estimating and Government contracting, as well as some training organizations experienced in parametric estimating. It also provides Internet addresses for other sites containing useful information on parametric estimating, Government contracting, and acquisition reform. Books and papers on quantitative business, cost, and statistical applications and techniques are also listed.

Professional Societies

The Association for the Advancement of Cost Engineering through Total Cost Management (AACE International)

Web Site: www.aacei.org

American Society of Professional Estimators (ASPE)

Web Site: www.aspenational.com

International Society of Parametric Analysts (ISPA)

Web Site: www.ispa-cost.org

National Contract Management Association (NCMA)

Web Site: www.ncmahq.org

Society of Cost Estimating and Analysis (SCEA)

Web Site: www.sceaonline.net

Colleges and Universities

Defense Acquisition University (DAU)

Web Site: www.dau.mil

Federal Acquisition Institute (FAI)

Web Site: accessible through SCEA website at www.sceaonline.net

Commercial Model Vendors

Galorath Incorporated

Web Site: www.galorath.com/SEER_tools.html

Mainstay Software Corporation

Web Site: www.mainstay.com

NASA/Air Force Cost Model

Web Site: nafcom.saic.com/

PRICE Systems, LLC

Web Site: www.pricesystems.com

Quantitative Software Management

Web Site: www.qsm.com

Other Web Sites

COCOMO Models

Web Site: sunset.usc.edu/research/cocomosuite/index.html

Contract Pricing Reference Guides

Web Site: www.acq.osd.mil/dp/cpf/pgv1_0/pgnew.html

Cost Estimating Resources

Web Site: www.jsc.nasa.gov/bu2/resources.html

Defense Acquisition Deskbook

Web Site: www.deskbook.dau.mil

DoD Acquisition Initiatives

Web Site: www.acq.osd.mil/ar/ar.htm

Dr. Edmund H. Conrow's CAIV Web Site

Web Site: www.caiv.com/index.htm

Information Systems Audit and Control Association (ISACA)

Web Site: www.isaca.org

Management Consulting and Research, Inc. (MCR)

Web Site: www.mcri.com

Software Engineering Institute (SEI)

Web Site: www.sei.cmu.edu

Software Engineering Laboratory (SEL)

Web Site: sel.gsfc.nasa.gov

Software Technology Support Center (STSC)

Web Site: www.stsc.hill.af.mil

Statistical Resources on the Internet

Web Site: www.lib.umich.edu/govdocs/stecon.html

Books on Quantitative Applications

Applied Regression Analysis by Norman R. Draper and Harry Smith

Basic Business Statistics: Concepts and Applications by Mark L. Berenson, David M. Levine, and Timothy C. Krehbiel

Basic Econometrics by Damodar Gujarati

Business Statistics: Decision Making with Data by Richard A. Johnson and Dean W. Wichern

Cost Estimating by Rodney D. Stewart

Econometric Models and Economic Forecasts by Robert S. Pindyck and Daniel L. Rubinfeld

Statistics for Business and Economics by David R. Anderson, Thomas A. Williams, and Dennis J. Sweeney

Statistical Techniques in Business and Economics by Douglas A. Lind, Robert Deward Mason, and William G. Marchal

Statistical Thinking and Data Analysis Methods for Managers by Wynn Anthony Abraonvic

APPENDIX E
Parametric Estimating Checklists

This appendix contains checklists of the main features which an acceptable parametric estimating system should consider or include, and which proposals based on parametric techniques should meet. Organizations and analysts who are implementing, or reviewing, parametric systems or proposal basis of estimates (BOEs) will find these checklists useful. In order to understand the various purposes of each of the listed points, read the checklists thoroughly before using them. For example, auditors could incorporate many points into their audit programs or when summarizing their findings. Contractors, on the other hand, can use the same points to assess the adequacy of their estimating procedures.

The checklists should be tailored as appropriate to each organization and application. Chapter 7, Government Compliance, provides additional information.

Checklist for Parametric Estimating Systems

General Policies and Procedures

Examples of Assessment Criteria	Yes	No
1. Does the contractor have adequate estimating policies and procedures that address proper implementation of parametric techniques?		
2. Do the established policies and procedures contain guidance for: • Appropriate use of parametric techniques? • Performing internal reviews to assess compliance with estimating policies and procedures, as well as consistency in estimating practices? • Complying with relevant Government procurement laws and regulations?		

Data Collection and Analysis

Examples of Assessment Criteria	Yes	No
1. Has the contractor established a process for collecting and analyzing relevant cost and noncost (e.g., technical information) data on a periodic basis?		
2. Does the contractor use a consistent format (e.g., work breakdown structure) for collecting cost and noncost data?		
3. Is data accumulated in a manner consistent with the contractor's estimating practices?		
4. Does the contractor identify and examine any data anomalies?		

Special Considerations Related to Cost Estimating Relationships (CERs)

Examples of Assessment Criteria	Yes	No
1. Does the contractor demonstrate the ability to establish valid CERs that are reliable and credible predictors of cost?		
2. Does the contractor's procedures for implementing valid CERs include: • Ensuring data relationships are logical? o Various alternatives (i.e., cost drivers) are considered o Cost drivers selected are good predictors of cost • Ensuring databases are credible? o Data are current, accurate, and complete o Data are appropriately analyzed and normalized o Any adjustments are logical, reasonable, and verifiable • Demonstration that the data relationships are strong? o Analytical tests are used to evaluate strengths • Evidence that non-quantitative information was also considered? o Materiality of cost being estimated o Quality of alternate estimating techniques		

Examples of Assessment Criteria	Yes	No
3. Does the contractor maintain documentation that includes mathematical formulas; independent and dependent variables; statistical analysis results (or other analytical test results if appropriate); and other data used to demonstrate the CERs validity?		
4. Are updating policies and procedures established that contain guidance related to: • Identifying significant data outside the normal update schedule for analysis, to determine if an out-of-period update is needed? • Identifying and incorporating savings associated with new ways of doing business (e.g., technology changes, single process initiatives, software process improvements)?		
5. Are monitoring processes established to ensure CERs remain reliable predictors?		
6. Are procedures in place to ensure that CERs are used appropriately in proposals?		
7. When Forward Pricing Rate Agreements (FPRAs) are established do they contain definitions of all variables; tracking requirements; and appropriate applications for the CERs?		

Special Considerations Related To Commercial Parametric Models

Examples of Assessment Criteria	Yes	No
1. Are estimating policies and procedures established that contain: • Brief description of the commercial model(s) used? • Guidelines for the calibration process? • Identification of the model's significant cost drivers and input parameters?		
2. Is the commercial model calibrated in accordance with policies and procedures, and does it include: • Use of historical data to the extent practical? • Demonstration that the data used in calibration were properly analyzed, normalized, and mapped? • Maintenance of adequate documentation, including		

Examples of Assessment Criteria	Yes	No
description of key inputs; input parameter values and associated rationale; calibration assumptions; results of interviews (e.g., names of people interviewed, dates, information obtained); cost history; and calibration results?		
3. Are adjustments made to the parameter values justified with appropriate assumptions and rationale?		
4. Is the model updated consistently with policies and procedures, and does it reflect: • Cut-off dates negotiated with the Contracting Officer for ongoing programs (if applicable)? • Adjustments for the effects of any significant events that may impact the estimate?		
5. Is the model validated to ensure it is a good predictor of costs?		
6. Are on-going monitoring processes established to maintain the reliability of the model to be a good predictor of costs?		
7. Are information system security controls established to maintain the integrity of the system?		

Special Considerations Related to Company Developed Models

Examples of Assessment Criteria	Yes	No
1. Did the contractor perform a cost-benefit analysis prior to implementing the company-developed model?		
2. Does the contractor demonstrate compliance with the applicable criteria described in the CERs category (above) and/or the commercial models category (above)?		
3. Have information system controls been established to monitor systems development and maintenance activities?		
4. Has the model been thoroughly tested to ensure it produces the expected results?		
5. Does the contractor develop and maintain documentation regarding the system description that includes information on the processing performed by the model; data processed by the model; reports generated by the model; and user instructions (including parameter values)?		

Examples of Assessment Criteria	Yes	No
6. If the company-developed model is complex, were people experienced with systems engineering and/or software engineering used to implement it?		

Other Considerations:

Examples of Assessment Criteria	Yes	No
1. Did the company perform adequate cost-benefit analyses prior to implementing parametric estimating techniques?		
2. Does the company compare its parametric estimates to prior actual costs or independent estimates to analyze significant differences?		
3. Do the parametric estimators: • Receive proper training on the techniques being used? • Have relevant experience? • Receive guidance regularly from their supervisors?		

Checklist for Proposals Based On Parametric Techniques

Examples of Assessment Criteria	Yes	No
1. Is the contractor's proposal compatible with the format described in the solicitation? (Note: This could include the format described in the Federal Acquisition Regulation (FAR) 15.408 (l-m), Table 15-2 or an alternate format as specified in the solicitation clause at FAR 52.215-20, "Requirements for Cost or Pricing Data or Information Other than Cost or Pricing Data.")		
2. Is the following information included with the contractor's submission? • Background on the parametric techniques used • Description of the data analysis and calibration processes • Information on the validation process • Application of the parametric technique to the current estimate, including details on the key inputs used, the contractor's rationale for the parameter values, and the		

Examples of Assessment Criteria	Yes	No
basis for any adjustments made to the model		
3. Did the contractor disclose any comparative estimates it used as sanity checks to demonstrate the reliability of the parametric estimates?		
4. When applicable, are any FPRAs for parametric techniques (e.g., CERs) described?		
5. When applicable, are relevant cut-off dates disclosed?		
6. Are parametric based estimates developed in accordance with policies and procedures?		
7. Are any deviations from the defined policies and procedures appropriately justified (including all assumptions and rationales)?		
8. Are parametric based estimates developed in accordance with the relevant Government procurement regulations?		
9. For estimates based on CERs: • Are the inputs for any costs being estimated within the appropriate range of the databases or extrapolated to reasonable amounts? • Are databases current, accurate, and complete (see 11 below)?		
10. For estimates based on calibrated and validated parametric models: • Did the contractor use the same version of the model to prepare the estimate that was previously calibrated and validated? • Did the contractor calibrate to the best and most relevant data? • Are the key cost drivers and associated parameter values identified? • Are the parameter values supported by appropriate rationale? • Are any adjustments made to the parameter values logical and reasonable? • Are databases current, accurate, and complete (see 11 below)?		

Examples of Assessment Criteria	Yes	No
11. Are databases appropriately updated? • Do any adjustments need to be made to incorporate savings associated with new ways of doing business? • Are any such adjustments supported with logical and reasonable rationale?		

Exhibit A: A Manager's Checklist for Validating Software Cost and Schedule Estimates

This checklist is designed to help managers assess the credibility of software cost and schedule estimates. It identifies seven issues to address and questions to ask when determining your willingness to accept and use a software estimate. Each question is associated with elements of evidence that, if present, support the credibility of the estimate.

Issue 1. Are the objectives of the estimate clear and correct?

Evidence of Credibility

- ❐ The objectives of the estimate are stated in writing.
- ❐ The life cycle to which the estimate applies is clearly defined.
- ❐ The tasks and activities included in (and excluded from) the estimate are clearly identified.
- ❐ The tasks and activities included in the estimate are consistent with the objectives of the estimate.

Issue 2. Has the task been appropriately sized?

Evidence of Credibility

- ❐ A structured process has been used to estimate and describe the size of the software product.
- ❐ A structured process has been used to estimate and describe the extent of reuse.
- ❐ The processes for estimating size and reuse are documented.
- ❐ The descriptions of size and reuse identify what is included in (and excluded from) the size and reuse measures used.
- ❐ The measures of reuse distinguish between code that will be modified and code that will be integrated as-is into the system.

- ❐ The definitions, measures, and rules used to describe size and reuse are consistent with the requirements (and calibrations) of the models used to estimate cost and schedule.
- ❐ The size estimate was checked by relating it to measured sizes of other software products or components.
- ❐ Testing its predictive capabilities against measured sizes of completed products checked the size estimating process.

Issue 3. Are the estimated costs and schedule consistent with demonstrated accomplishments on other projects?

Evidence of Credibility

- ❐ The organization has a structured process for relating estimates to actual costs and schedules of completed work.
  - ❐ The process is documented.
  - ❐ The process was followed.
- ❐ The cost and schedule models that were used have been calibrated to relevant historical data. (Models of some sort are needed to provide consistent rules for extrapolating from previous experience.)
- ❐ The cost and schedule models quantify demonstrated organizational performance in ways that normalize for differences among software products and projects. (So that a simple, non-normalized, lines-of-code per staff month extrapolation is not the basis for the estimate.)
- ❐ The consistency achieved when fitting the cost and schedule models to historical data has been measured and reported.
- ❐ The values used for cost and schedule model parameters appear valid when compared to values that fit the models well to past projects.
- ❐ The calibration of cost and schedule models was done with the same versions of the models that were used to prepare the estimate.
- ❐ The methods used to account for reuse recognize that reuse is not free. (The estimate accounts for activities such as interface design, modification, integration, testing, and documentation that are associated with effective reuse.)
- ❐ Extrapolations from past projects account for differences in application technology. (For example, data from projects that implemented traditional mainframe applications require adjustments if used as a basis for estimating client-server implementation. Some cost models provide capabilities for this, others do not.)
- ❐ Extrapolations from past projects account for observed, long-term trends in software technology improvement. (Although some cost models

attempt this internally, the best methods are usually based on extrapolating measured trends in calibrated organizational performance.)

- ❐ Extrapolations from past projects account for the effects of introducing new software technology or processes. (Introducing a new technology or process can initially reduce an organization's productivity).
- ❐ Workflow schematics have been used to evaluate how this project is similar to (and how it differs from) projects used to characterize the organization's past performance.

Issue 4. Have the factors that affect the estimate been identified and explained?

Evidence of Credibility

- ❐ A written summary of parameter values and their rationales accompanies the estimate.
- ❐ Assumptions have been identified and explained.
- ❐ A structured process such as a template or format has been used to ensure that key factors have not been overlooked.
- ❐ Uncertainties in parameter values have been identified and quantified.
- ❐ A risk analysis has been performed, and risks that affect cost or schedule have been identified and documented. (Elements addressed include issues such as probability of occurrence, effects on parameter values, cost impacts, schedule impacts, and interactions with other organizations.)

Issue 5. Have steps been taken to ensure the integrity of the estimating process?

Evidence of Credibility

- ❐ Management reviewed and agreed to the values for all descriptive parameters before costs were estimated.
- ❐ Adjustments to parameter values to meet a desired cost or schedule have been documented.
- ❐ If a dictated schedule has been imposed, the estimate is accompanied by an estimate of:
  - ❐ The normal schedule;
  - ❐ The additional expenditures required meeting the dictated schedule.
- ❐ Adjustments to parameter values to meet a desired cost or schedule are accompanied by management action that makes the values realistic.
- ❐ More than one cost model or estimating approach has been used, and the differences in results have been analyzed and explained.

- ❐ People from related but different projects or disciplines were involved in preparing the estimate.
- ❐ At least one member of the estimating team is an experienced estimator, trained in the cost models that were used.
- ❐ Estimators independent of the performing organization concur with the reasonableness of the parameter values and estimating methodology.
- ❐ The groups that will be doing the work accept the estimate as an achievable target.
- ❐ Memorandums of agreement have been completed and signed with the organizations whose contributions affect cost or schedule.

Issue 6. Is the organization's historical evidence capable of supporting a reliable estimate?

Evidence of Credibility

- ❐ The estimating organization has a method for organizing and retaining information on completed projects (a historical database).
- ❐ The database contains a useful set of completed projects.
- ❐ Elements included in (and excluded from) the effort, cost, schedule, size, and reuse measures in the database are clearly identified. (See, for example, the SEI checklists for defining effort, schedule, and size measures.)
- ❐ Schedule milestones (start and finish dates) are described in terms of criteria for initiation or completion, so that work accomplished between milestones is clearly bounded.
- ❐ Records for completed projects indicate whether or not unpaid overtime was used.
- ❐ Unpaid overtime, if used, has been quantified, so that recorded data provide a valid basis for estimating future effort.
- ❐ Cost models that were used for estimating have been used also to provide consistent frameworks for recording historical data. (This helps ensure that comparable terms and parameters are used across all projects, and that recorded data are suitable for use in the estimating models.)
- ❐ The data in the historical database have been examined to identify inconsistencies, and anomalies have been corrected or explained. (This is best done with the same cost models that are used for estimating.)
- ❐ The organization has a structured process for capturing effort and cost data from ongoing projects.
- ❐ The producing organization holds postmortems at the completion of projects.
  - ❐ To ensure that recorded data are valid.

   - To ensure that events that affected costs or schedules get recorded and described while they are still fresh in people's minds.
   - Information on completed projects includes:
   - The life-cycle model used, together with the portion covered by the recorded cost and schedule.
   - Actual (measured) size, cost, and schedule.
   - The actual staffing profile.
   - An estimate at completion, together with the values for cost model parameters that map the estimate to the actual cost and schedule.
   - A work breakdown structure or alternative description of the tasks included in the recorded cost.
   - A workflow schematic that illustrates the software process used.
   - Nonlabor costs.
   - Management costs.
   - A summary or list of significant deliverables (software and documentation) produced by the project.
   - A summary of any unusual issues that affected cost or schedule.
- Evolution in the organization's workflow schematics shows steady improvement in the understanding and measurement of its software processes.

Issue 7. Has the situation changed since the estimate was prepared?

Evidence of Credibility

- The estimate has not been invalidated by recent events, changing requirements, or management action (or inaction).
- The estimate is being used as the basis for assigning resources, deploying schedules, and making commitments.
- The estimate is the current baseline for project tracking and oversight.

APPENDIX F

Memorandum of Understanding for Parametric Models

The following is a list of recommended items to include in a memorandum of understanding (MOU) useful for parametric models for forward pricing.

1. Opening statement defining the purpose of the MOU, with a focus towards ensuring compliance with TINA.
2. Parameters defining the currency of data, with care not to improperly waive or limit application of TINA.
3. List limiting usage of the MOU to specific sets of CERs, models, or model parts, as well as a clearly defined list of proposals covered, or not covered.
4. Provision permitting modification of the MOU by mutual agreement of the parties, as well as termination by either party.
5. Period of applicability.
6. Detailed description of the input data, defining said data and the cut-off date for updating each data element, defined when feasible, as specific reports, systems, or data files thereby avoiding general statements, such as "update labor monthly." Note: Cut-off dates for parametric databases may be monthly, quarterly, or even annually, if such time periods are current and complete as intended by TINA. .
7. Notification of unusual events, past or projected, e.g., changes in methodologies, and data sources, reorganizations, that may invalidate terms of the MOU, as well as impact specific proposals in excess of given thresholds.
8. Provision as to how and when the model or CERs will be validated.
9. Provision addressing audit access or review procedures germane to company records to facilitate efforts not directly required by TINA, but necessary under contract terms.
10. Procedures to access the model for review, test, validation, or supporting negotiation positions in order to validate compliance with, but not required by TINA..
11. Notification to the government of the most current, yet out-of-cycle data that significantly impacts proposal costs as defined in either absolute and/or percentage of difference amounts.

APPENDIX G

Parametric Cost Estimating Initiative Closure Report

Background

Origins

Parametric estimating is a technique that uses validated relationships between a project's known technical, programmatic, and cost characteristics and known historical resources consumed during the development, manufacture, maintenance, and/or modification of an end item. Industry and Government practitioners have used parametric techniques to perform a variety of applications, such as independent cost estimates and trade studies. However, Industry's use of parametric techniques as a basis of estimate (BOE) on proposals submitted to the Government or higher tier contractors was limited. This was a result of:

- A lack of awareness or understanding of parametrics, both within Industry and the Government acquisition community;
- Perceptions that regulatory barriers existed; and
- Limited examples of actual proposal applications.

Industry and Government parametric practitioners asserted that proposal preparation, evaluation, and negotiation costs and cycle time could be reduced considerably through the increased use of parametric estimating. These practitioners also stated that these benefits could be achieved while maintaining or improving the quality of the estimates produced.

Objectives

In December 1995, the Commander of the Defense Contract Management Agency (DCMA) and the Director of the Defense Contract Audit Agency (DCAA) formally sponsored a Reinvention Laboratory to evaluate the use of parametric estimating on proposals. Thirteen locations established integrated product teams (IPTs) to test the use of these techniques on proposals. These locations included divisions of Boeing (four locations participated), General Electric, Lockheed Martin (three locations participated), Motorola, Northrop Grumman (two locations participated), and Raytheon (two locations participated).

The Parametric Estimating Reinvention Laboratory objectives included identifying:

- Parametric applications to use as a BOE;
- Specific barriers precluding the expanded use of parametric techniques and procedures used to address these issues;
- Requirements of a valid parametric estimating system;
- Resources needed to support the implementation of parametrics; and
- Benefits that could be achieved using parametric techniques as a BOE.

In addition, the Laboratory objectives included developing case studies based on the IPTs' best practices and lessons learned for use in (i) revising the Parametric Estimating Handbook, and (ii) establishing a formal training course.

Results

To address the Laboratory objectives, the participants implemented a variety of parametric applications including cost estimating relationships (CERs), company-developed models, and commercial models. The Industry participants were able to negotiate contracts with their customers for proposals based on parametric techniques. In several of the test cases proposal preparation, evaluation, and negotiation costs and cycle time were reduced between 50% and 80% when parametric techniques were used as the primary BOE. In addition, the IPTs found that the accuracy of the estimates was maintained or improved because of the increased use of historical data. The Laboratory results also demonstrated that when properly implemented and used, parametric estimating complies with the Government procurement laws and regulations including the Truth in Negotiations Act (TINA), the Federal Acquisition Regulation (FAR), and Cost Accounting Standards (CAS). Case studies and examples were developed based on the IPTs' best practices and lessons learned, and they are incorporated into this Handbook.

While the above objectives were accomplished, two primary obstacles were identified that hindered the use of parametrics as a BOE on proposals. First, many contractors did not possess the necessary historical data to support parametric estimating. Second, there was uncertainty and/or unfamiliarity throughout Industry and the Government regarding the validity of parametric estimating. While these challenges were significant, the Reinvention Laboratory found they could be tackled by adopting the best practices defined in the Handbook.

The next section of this report provides information on how organizations can determine if parametric estimating would be beneficial to them. Also identified are key implementation practices.

Assessing the Feasibility of Parametric Estimating

Contractors generally implement parametric techniques to streamline costs and cycle time associated with Government contracting practices. Further, some contractors implement parametrics to improve the quality of their estimating practices by increasing the use of historical data. Whatever the reason(s), it is important for companies to address the feasibility of using parametric techniques prior to their implementation. In determining feasibility, some of the critical issues that organizations should assess are:

- Availability of relevant historical data;
- Reliability of current estimating techniques versus parametrics;
- Costs versus benefits;
- Industry and Government support.

These issues are described below. In addition, this Handbook provides detailed implementation guidance, including specific examples on these topics.

Availability of Relevant Historical Data

Since parametric techniques are based on historical information, the availability of current, relevant data is fundamental. The Laboratory demonstrated that companies should commit sufficient resources to establish credible databases that consist of relevant historical cost and noncost (technical) information. Several of the companies participating in the Laboratory used Information Technology to facilitate the data collection processes by establishing systems to collect and document program history in a consistent format. Industry's use of Enterprise Resource Planning (ERP) Systems[1] should also help to facilitate data collection activities and expand the use of parametrics.

Reliability of Current Estimating Techniques Versus Parametrics

Contractors should demonstrate that the parametric techniques used will be good predictors of cost and be able to produce estimates that are within a reasonable degree of accuracy. Contractors will also need to show that the parametric techniques implemented will be as credible as current estimating techniques. As stated, the Laboratory results proved that the integrity of estimates should be maintained or improved due to the increased use of historical data.

[1] ERP Systems are software applications (generally commercial products) that are integrated throughout an enterprise. These integrated applications generally include modules for engineering management, financial management, resource planning, management operations, and business management. When implemented properly, ERP Systems can have a significant impact on the working patterns of an organization.

Cost Versus Benefits

Contractors should perform a cost-benefit analysis to determine if the expected benefits (e.g., reduced costs and cycle time) are achievable. Proper implementation generally requires a significant start-up investment of resources to (i) support data collection and analysis activities, (ii) develop a valid parametric estimating system, and (iii) provide training to Industry and/or Government people who will be involved with proposals based on these techniques. When complex models are being contemplated, companies must assess the cost to procure a commercial off the shelf (COTS) model or to develop and maintain a company-developed model. Clearly, implementation costs will vary based on the type of parametric application desired (i.e., CERs or complex models). When assessing the potential use, other parametric applications should be considered. For example, Laboratory participants that implemented complex models were able to use these techniques to support other activities such as subcontract price/cost analysis, cost as an independent variable (CAIV) analysis, and certain contract risk management actions. Consequently, other uses of these techniques will help to increase a company's return-on-investment and their effects should be included in the cost-benefit analysis.

Industry and Government Support

It is essential for contractors to obtain the full support and commitment from all the people who will be affected by these new methodologies prior to implementation. For example, support is needed from company representatives involved in activities related to program management, finance, engineering, and manufacturing. In addition, support is needed from the company's primary customers (including higher tier contractors and Government procurement activities) and local representatives from DCMA and DCAA.

It is also critical that senior management from affected Industry and Government organizations be willing to accept the challenge to either implement and/or support the use of these new techniques. The Laboratory proved that senior management is most proactive when they participate in establishing and monitoring specific implementation goals. Naturally, their leadership is also needed to commit proper resources to demonstrate to their respective working forces their determination to use and/or support the use of parametrics as a BOE in the immediate future.

Implementing Parametric Estimating Techniques

Based on the above criteria, if a company has determined that parametric estimating will be beneficial, implementation should proceed. The Laboratory accomplishments resulted in several best practices companies should use to achieve their goals, including:

- Establishing an implementation team,
- Providing and participating in joint training;

- Developing parametric estimating system policies and procedures.

These practices are also highlighted below. Note that the Parametric Estimating Handbook provides detailed implementation guidance and specific examples on these topics.

Establishing an Implementation Team

Proper implementation of parametric techniques benefits from the use of Joint Industry/Government teams. The companies that participated in the Reinvention Laboratory established a two-tier structure consisting of Management Councils and IPTs. Both of these groups were comprised of representatives from the company, primary customer(s), DCMA, and DCAA. The Management Councils consisted of senior level people who were responsible for committing the necessary resources, providing guidance, managing the staff activities, and helping to resolve any issues. The IPTs, however, were made up of working level people who were involved with the implementation or evaluation of these techniques.

The use of a teaming process provided the Industry and Government participants visibility into the parametric estimating process. For example, it helped those people previously unfamiliar, to gain trust in and support for the parametric estimating technique(s). Teaming also provided contractors with an opportunity to address up-front, the concerns of those who would be affected by the new estimating process.

Providing and Participating in Joint Team Training

A key element to the IPTs implementation of parametric estimating related to the delivery of joint team training. During the Laboratory, the Defense Acquisition University delivered pilot training to the IPTs. This training provided the IPTs insight into the criteria of a good parametric estimating system. In addition, the training served as a mechanism to facilitate teambuilding and it helped the IPTs to develop their specific implementation objectives and action plans. In addition to the DAU sponsored training, most of the contractors educated their IPT members in other areas, including:

- The specific model(s) being implemented and used;
- Relevant product history;
- the company's cost accounting system.

Joint IPT training was a significant contributor to the Laboratory success and should be continued if parametric estimating will be institutionalized throughout the acquisition community. Information related to outside training available on parametric cost estimating can be obtained from either the International Society of Parametric Analysts (ISPA) or the Society of Cost Estimating and Analysis (SCEA).

Developing Parametric Estimating System Policies and Procedures

A contractor's implementation of parametric techniques should focus on the development of a valid estimating system. A valid estimating system should include:

- Demonstration that the parametric techniques are credible estimating tools (i.e., good predictors of cost),
- Data collection and analysis,
- Calibration[2] activities that have been thoroughly performed and documented,
- Personnel with proper experience and relevant training, and
- Formal policies and procedures that provide guidance on these topics as well as on the use of parametrics as a BOE.

A contractor's parametric estimating procedures should address any regulatory requirements, including updating procedures. For example, contractors should establish procedures to ensure their estimates are based on the most current and relevant data, and that they reflect current processes (e.g., savings associated with issues, such as acquisition streamlining, use of IPT processes, and manufacturing and/or software process improvements).

Future of Parametric Estimating

While the Laboratory results showed that proposal preparation, evaluation, and negotiations costs and cycle time can be reduced up to 80% when parametric techniques are used as the primary BOE, continued implementation will require organizations to address the following challenges:

- Industry should demonstrate their true commitment to using parametric techniques as a BOE by establishing formal data collection systems.
- Contractors should encourage their major subcontractors to implement and use properly calibrated and validated parametric techniques.
- Industry and Government should work together to identify opportunities for using parametric techniques.

The Laboratory demonstrated that parametric training should be offered to new IPTs involved in the implementation or evaluation of parametric techniques. Many people in the acquisition community are often skeptical about parametrics

[2] Calibration is a process of adjusting a commercial parametric model to a specific contractor's historical cost experience and business culture. Calibration is accomplished by calculating adjustment factor(s) that compensate for differences between an organization's historical costs and the costs predicted by the cost model that are based on default values. The proper calibration of a commercial model has a significant impact on the contractor's ability to produce accurate estimates. Consequently, commercial parametric models should be calibrated before they are used as a BOE for proposals submitted to the Government or higher tier contractors.

because they are not familiar with estimating techniques that utilize cost estimating models. While this Handbook should help to facilitate awareness, training should help facilitate the understanding of the proper use of parametric techniques.

APPENDIX H

Space Systems Cost Analysis Group Risk Summary

Why Probability Distributions and Not Point Estimates

The following text is an extract from the SSCAG Space Systems *Cost Risk Handbook*, 2005.

> There is a tendency by the cost modeling community to underestimate the true cost of space programs simply due to methodological issues. "There are many other reasons for this phenomenon, but one of the main reasons cost estimators underestimate acquisition costs is due to the fundamental inability to predict the future. Since it is impossible to make accurate predictions, the cost community has relied heavily on the development of cost estimating relationships (CERs), based on historical cost data, for the purpose of making statistically-based, educated guesses about the cost of systems that have yet to be built. Moreover, the use of CERs requires exact knowledge of the future system's design – even if it hasn't been designed yet! And while CERs have served the cost community well, they are fundamentally a regression curve through a subset of historical cost data, and if not used correctly, or if the wrong assumptions are made, will produce misleading results."
>
> A typical CER is based on historical cost data. The regression "curve that best fits the data has the…feature of increasing as the cost driver increases. Thus, if the cost driver represents a key physical or performance parameter such as weight or power, then as weight or power increases, the cost estimate also increases. Therefore, one can use this CER by evaluating the function at any value of the cost driver, and it will provide an estimate of the cost of a similar system having that property."
>
> "Unfortunately this CER also has some drawbacks. First, while the regression curve tracks with the data, it doesn't correctly estimate any of them. Each data point falls some distance away from the regression curve. Second, some of the data points are lower than the curve, and some are substantially higher. So, had a

cost estimator used this CER to estimate the cost of one of these data points, he would have missed it.

Moreover, it has been shown that "historical space cost data tends to be distributed such that the errors are proportional, that is, they increase as the cost driver increases, and that the errors tend to be skewed toward the high side. The implication is that gross underestimates are more likely than gross overestimates. There is a high likelihood that the true cost may be significantly larger that the mean. This argues for the necessity of accounting for the spread, or variability, of the data when using the CER to estimate cost."

"All of this points to what is perhaps the biggest reason that we consistently underestimate the cost of space systems … cost estimates are *really* probability distributions and *not deterministic*. In a perfect world, cost analysts should present cost estimates as probability distributions, and acquisition decision-makers should then choose their estimate, or budget, by balancing that choice against the risk of a budget overrun. Naturally, the higher the cost, or budget, estimate, the lower the probability of a budget overrun. But the prevailing practice is that cost estimators report, say, the 50th percentile of the probability distribution as *the cost estimate*, and decision-makers choose to budget at a value that is even less than that."

APPENDIX I

Space System Cost Analysis Group, Nonrecurring and Recurring Cost Definitions and Cost Accounting Guidelines

SPACE SYSTEM COST ANALYSIS GROUP

NONRECURRING AND RECURRING COST DEFINITIONS

AND

COST ACCOUNTING GUIDELINES

January 7, 2006

Table Of Contents

SSCAG NONRECURRING AND RECURRING COST DEFINITIONS AND COST ACCOUNTING GUIDELINES

January 7, 2006

1. Introduction and Background

This document is a product of the Space System Cost Analysis Group (SSCAG), a joint government-industry organization devoted to advancing the art of estimating space system costs. It was developed with the intent of bringing more consistency into the estimating processes of SSCAG member organizations and, ultimately, adoption as an industry standard on a broader basis.

Space system development costs are generally nonrecurring in nature, while production costs recur when system components (spacecraft and ground stations or portions thereof) are produced. Estimating nonrecurring (NR) and recurring (Rec, or R) costs separately is instrumental in predicting the costs of future space system acquisitions. Naturally, segregation of actual NR and Rec costs from completed, and even partially complete, acquisition contracts is needed to support these predictions.

Although both government and industry have used procedures and definitions to segregate nonrecurring and recurring costs for decades, no industry standard has evolved from these practices. There has been general agreement on many basic criteria for segregating NR and Rec costs, but significant differences have occurred from case to case. Also, in many instances, proposed and/or actual NR and Rec costs for government-funded systems have not been segregated because the government did not require it.

A survey of SSCAG members was conducted in 2001 and early 2002 to shed more light on how contractors and government agencies deal with nonrecurring and recurring acquisition costs. The results of the survey revealed a need for standard definitions. Some organizations had no definitions while other organizations had only top level, summary definitions.

The most common reason cited (by both government and industry) for not recording NR and Rec costs separately has been the additional expense to do so. In some cases, NR and Rec actual costs have not been segregated because the benefits (in predicting costs for follow-on acquisitions) were either not acknowledged or they were not considered important enough to justify the additional costs associated with the segregation. However, the problem of establishing simple and unambiguous implementation procedures has probably been a factor as well.

SSCAG has expanded and refined definitions originally developed in 2002 to address most costs associated with space systems, yielding the current set of definitions. Further expansion is anticipated in the future – primarily for launch systems.

The definitions represent a consensus of the SSCAG member organizations – both government and industry – through their SSCAG representatives, who participated in developing the definitions. Although no formal commitments to use the definitions have been made, SSCAG fully expects such commitments from key government agencies and contractors responsible for space system acquisition and operation.

2. Purposes and Objectives

The purpose of these nonrecurring and recurring cost definitions is to provide SSCAG member organizations with commonly accepted definitions of NR and Rec costs as they apply to space system acquisition and operation. The adoption of these definitions should promote more universal, consistent NR and Rec cost segregation. This, in turn, should assist the member organizations in producing more realistic space system cost estimates and enhance comparisons between estimates from different sources.

The cost accounting guidelines are intended to assist contractors in establishing appropriate cost accounting practices for recording NR and Rec costs. Removal of roadblocks to the mechanics of cost segregation should reduce the incremental cost of recording NR and Rec costs separately.

SSCAG's objective in developing the NR and Rec definitions is to strike a compromise between the "purest" definitions possible and practical considerations. Specifically, compromises were made when relatively small costs were involved and segregating them would be either difficult or not worth the effort. However, the definitions were not compromised if the costs involved could be relatively large – even though they might be small in many cases. Classification of redesign, rework and retest to correct design deficiencies as nonrecurring is the most notable instance of not compromising the definitions. Cases where the definitions represent a compromise are identified in Section 5 below.

One of the most significant decisions made by SSCAG was to define the costs of virtually all hardware units installed in ground stations as recurring (while their development is nonrecurring). The purpose of this approach is to yield actual costs which can be used as the basis for estimating the costs of replicating the ground stations. When ground station support hardware is used to support both system development and (later) system operations and support, SSCAG opted to define such hardware as recurring (with minor exceptions). The overriding consideration was that of establishing the most complete set of costs needed to provide a basis for estimating the cost of additional, replicated ground stations. For the same reason, costs for installation of previously-developed software and the costs for commercial off-the-shelf (COTS) software installed in ground stations are also defined as recurring.

The definitions are not tied to budget appropriation codes. Common defense appropriation codes, such as the OSD 3600 code for RDT&E and the 3020 code for procurement, usually entail both nonrecurring and recurring efforts for space systems. No attempt was made to relate the SSCAG definitions to appropriation codes because the primary purpose of the definitions is to support cost estimating, as opposed to budgeting.

SSCAG views the NR and Rec definitions as a goal to strive for and acknowledges that they will not necessarily be followed in their entirety on any given space system acquisition. The degree to which they will be followed will depend primarily on customer guidance and contractor response to that guidance. SSCAG urges all member organizations to adopt the definitions and practice the recommended cost accounting approaches to more realistic record NR and Rec costs.

3. General Definitions

General definitions of space system nonrecurring and recurring costs are highlighted in Table 3-1.

Acquisition costs for development of the space system are classified as nonrecurring at all levels of indenture. This includes initial development, upgrades, and improvements to hardware, software and facilities that make up the system. Thus, requirements definition, design, engineering analyses, development hardware and software, development testing, and development support hardware and software are nonrecurring efforts.

Acquisition costs for production of operational spacecraft, for ground stations and components thereof, for spares, and for replicated spacecraft components used in ground testing and simulators are classified as recurring. Costs for installation of previously developed software in ground stations are recurring. Similarly, costs for acquisition and installation of commercial off-the-shelf software (COTS) for operational use in ground stations are recurring.

Operations and Support (O&S) costs for development of concepts, procedures, training materials and simulators are classified as nonrecurring. Costs for system operation and maintenance, including training, are recurring.

Table 3-1 General Nonrecurring and Recurring Cost Definitions

Nonrecurring Cost	Initial development of spacecraft, ground stations and other system segments/elements/interfaces Support equipment and software Upgrades and improvements Applies to hardware, software, facilities and O&S
Recurring Cost	Spacecraft production, integration and test Ground hardware production and installation Installation of developed software in ground stations Replicated space hardware in ground stations Spares for spacecraft and ground stations Test equipment, software and procedure maintenance System operation and maintenance

These general definitions are not easily interpreted in many cases. Therefore, they are expanded in Section 5 to draw a more explicit line between NR and Rec costs.

The term “spacecraft” is meant to include a spacecraft bus and all payloads. Other terms with generally the same meaning are “space vehicle” and “satellite”. For manned spacecraft, life support systems, docking subsystems, manipulating arms and the like are viewed as part of the spacecraft. Launch vehicle adapters and multi-spacecraft “dispensers”, when combined with the spacecraft, represent “space hardware”. The “space segment” of a space system is composed of one or more spacecraft, which may be different in configuration and function.

The "ground segment" of a space system is made up of one or more ground stations or portions thereof. Most ground stations include mission planning, space-ground communication, spacecraft command and control, and mission data processing functions. Other ground-based facilities, equipment and software may also be included in a ground segment or in other system segments (e.g., for data fusion and product dissemination).

Launch vehicle fairings are usually viewed as part of the "launch segment", which also includes launch vehicles, launch segment integration (LSI), launch bases and launch base support services (as opposed to launch support provided by spacecraft producers). Launch vehicle adapters might be either part of the space segment or launch segment. In some cases, adapters are provided by spacecraft contractors. In other cases, they are purchased along with launch vehicles.

The term "cost account" takes on a specific meaning in this document, driven by its typical use in earned value reporting. A cost account (CA) contains the accumulation of costs and labor hours from lower-level accounting entities such as "job numbers" (J/Ns), "work orders" (W/Os) and "cost charge numbers" (CCNs), depending on the contractor's accounting system.). For convenience, "job number", or "J/N", is used to describe the true lowest level of accounting indenture. J/Ns collect labor, parts, materials, subcontracts or combinations of these costs. Thus, the CA is the lowest level of work indenture reported separately in earned value reports, but not the lowest level of cost accounting indenture. Cost accounts are accumulated into Contract Work Breakdown Structure (CWBS) elements, which in turn are typically combined at the highest CWBS level into Contract Line Items (CLINs).

4. Applicability

These definitions and guidelines are intended for application to system acquisition, and to operation and support throughout the operational life of the system. Acquisition refers to full-scale system development and production of both spacecraft and ground stations and other ground resources considered to be an integral part of the space system. System operation and support applies primarily to ground stations and other ground resources, but it also includes support for launch and mission operations by spacecraft producers (although the latter are usually embedded in acquisition contracts or CLINs). O&S for space systems includes operation of spacecraft and ground segment subsystems/functions, planning and executing mission tasking, and maintaining all system assets. Evolutionary enhancements to ground segment software performed by O&S personnel and replacement of obsolete ground hardware (recapitalization) are generally accepted as O&S, but major ground station upgrades are categorized as acquisition.

Launch vehicle and launch support costs are not specifically addressed, although the definitions for space hardware, software and SEIT/PM (System Engineering, Integration and Test and Program Management) could be applied to launch vehicles. Similarly, the ground hardware, software, SEIT/PM and O&S definitions could be applied to launch bases and launch-related O&S.

Terrestrial communications (TCOM) is also not addressed, but the ground hardware, software, SEIT/PM and O&S definitions could be applied to TCOM.

Dissemination of system products and data is only addressed to the extent that the required resources are defined to be included within the system boundaries (i.e., ground stations, system O&S).

Costs for system concept development and technology development prior to full-scale development (i.e., prior to acquisition contract Authority to Proceed, or ATP) are usually considered to be nonrecurring. They are not addressed further in the definitions.

The definitions are intended for application to all levels of integration, or indenture, within the space system:

- Parts, materials and labor
- Hardware components, subassemblies, major assemblies and end items (“boxes”)
- Software Computer Software Configuration Items (CSCIs) and Components (CSCs)
- Subsystems, system segments, and the system as a whole

The definitions apply to all contractor costs contributing to space system acquisition and O&S. The ultimate goal of these definitions is to separate all contracted space system acquisition and O&S costs into either the nonrecurring or recurring category. Launch and terrestrial communications definitions are needed to achieve this goal; their development is planned as a future SSCAG task. Effort provided by government employees is usually not quantified in terms of nonrecurring and recurring components, but the definitions can be used for this purpose.

5. Expanded Nonrecurring and Recurring Cost Definitions

The expanded definitions are organized into space hardware, ground station hardware, software, SEIT/PM and O&S categories. The software definitions apply to flight software, ground station operational software and ground test software for spacecraft development and production. Ground support equipment (GSE) is addressed within the space and ground station hardware sections below. GSE is also addressed in the SEIT/PM section, although it is often viewed as separate from integration and test for cost collection and estimating purposes.

The SEIT/PM definitions apply primarily to the segment (space, ground) and system levels. The term "system" as used here refers to, as a minimum, the combination of the space segment (i.e., spacecraft and their associated SEIT/PM) and the command and control elements of one or more ground stations (including associated SEIT/PM). However, the term in its broader context may also include other ground station elements (communications, mission data processing, facilities, etc.), launch segments (primarily launch vehicles and launch segment integration), the overall collection of system segments and possibly other entities/costs.

SEIT/PM definitions may also be applied to major elements within space and ground segments, such as: the spacecraft bus, payloads, subsystems, ground processing groups containing both hardware and software, software CSCIs and higher-level software entities/functional groups. In this case, the "System" in "System Engineering" is understood to mean the element it refers to, as opposed to the system in its entirety. The same applies to Integration and Test (I&T) and "Program" Management. (Program Management is usually only the highest level of management within a contract, while lower-level management efforts are usually referred to by terms such as "Spacecraft Bus Management", "Payload Management" and "Ground Command and Control Subsystem Management".)

Generally, management, engineering, integration (assembly), and testing devoted to hardware end-items (i.e., hardware assemblies or "boxes") are not viewed as SEIT/PM efforts but rather "box-level" management, engineering, and I&T, usually embedded in the total box costs. Hardware end-items are viewed as being made up of lower-level elements called "components", "modules" or "subassemblies". Similarly, software end-items, or Computer Software Configuration Items (CSCIs), may be viewed as being made up of software components (CSCs) and modules. Some amount of System Engineering is usually embedded within CSCI developments even though they are end-items. This form of System Engineering may be more strictly referred to as "Software System Engineering" or, more simply, as "Software Engineering".

5.1. Space Hardware

This section provides more explicit definition of nonrecurring and recurring costs for space hardware. First, the various types of space hardware units are identified and described, leading to hardware unit definitions which, in turn, are used in defining nonrecurring and recurring costs.

Hardware Unit Definitions

The cost descriptions in this section make reference to different types of hardware units: breadboard, brassboard, engineering, traditional qualification, protoqual, protoflight, flight and spare. Of these terms, “traditional qualification”, “protoqual”, and “protoflight” have had different meanings to different space system estimating organizations. The following definitions have been agreed upon within the SSCAG working groups and may be at odds with current definitions of member organizations. However, each organization should be able to translate their nomenclature into these SSCAG definitions.

The term “traditional qualification” means a unit that is primarily for design qualification. These units are subjected to full qualification level and/or life testing, which may cause some amount of degradation or damage. These units may also be referred to as simply “qualification” or “qual” units, and are not usually intended to be a part of operational spacecraft, although in many cases they are refurbished and installed on follow-on spacecraft. For the SSCAG definitions, the term “traditional” is dropped, and these units are referred to as “qualification” units. The term “qual” is retained as an abbreviation .

Some qualification units may be incomplete. For example, solar array qual units typically have only a small portion of the full solar cell population. Similarly, electronic qual units, when internally redundant, may have only one “leg” or “string” of the redundant componentry. However, qual units are necessarily composed of flight quality hardware, to test the viability of designs, parts, materials and fabrication procedures in the operational environment.

Hardware units that are subjected to less severe testing levels than those for qualification units are usually referred to as “protoqual” or “protoflight” units. These units are intended to become part of operational spacecraft, usually on “first of a kind” vehicles (however, they may be intended for follow-on vehicles instead). The testing levels for these units are referred to as “protoqual levels” or “protoflight levels”, as opposed to “full qualification levels” or just “qual levels”. The protoflight/protoqual test levels are such that they are not expected to cause any damage -- yet be sufficiently severe to gain confidence in the operational viabilty of the hardware. For purposes of the SSCAG NR and Rec definitions, “protoqual” and “protoflight” are used synomonously, even though some member organizations may draw distinctions between the two. “Protoflight” has been arbitrarily selected vice “protoqual” for use in the definitions.

The term “flight unit” refers to a unit that is exposed to acceptance level testing, as opposed to qualification and protoflight levels. Flight units are typically produced and tested after one protoflight unit is produced and tested. However, in some cases, more than one protoflight unit may be produced (as extra insurance that the design is adequate). Protoflight and flight units are physically identical, with the possible exception of minor design “tweaks”. The only significant difference between the two is the testing levels to which they are subjected.

In some cases, “engineering” units are referred to as “engineering development units” (EDUs) or simply “development units” are produced to demonstrate the viability of the basic hardware design. These units typically have the same “form, fit and function” as protoflight and flight units, but need not be built to the same standards. They may not have flight-quality parts and, like qualification units, may not have a complete set of internal components.

Breadboard and brassboard units are typically representative of future engineering, qual, and flight units with regard to functions, but are often very different in form and fit. These units may be characterized more as “laboratory models” while the other units are “flight-like”. Breadboard and brassboard units often exclude functions not critical to establishing design feasibility.

Space Hardware Nonrecurring Costs

Space hardware nonrecurring costs, listed in Table 5-1 on the next page, include all development efforts, parts and materials needed to qualify component, end-item, subsystem and spacecraft designs, including:

- Requirements definition, design (preliminary and detailed) and engineering analyses
- Tooling, fixtures and procedures for manufacturing and assembling hardware units
- Manufacture, assembly and testing of development hardware units: breadboard, brassboard, engineering, qual and life test units
- Parts and materials used in development units
- Development and production (or acquisition) of ground support and handling equipment, test equipment and hardware portions of simulators/stimulators (referred to subsequently as “ground support equipment”)
- Development test planning, conduct, tear down, analysis and documentation
- Engineering support for development unit manufacture and test
- Development of all test procedures – for both development and production acceptance testing
- Redesign, rework and retest of engineering, qualification, protoflight, flight and spare units, if necessary, to correct design problems
- Refurbishment and retest of qualification and protoflight units, typically for subsequent use on operational spacecraft
- Maintenance of designs; fabrication and assembly procedures; and acceptance test procedures
- Maintenance of ground support equipment until first spacecraft launch

Some of the efforts/costs above apply primarily to hardware assemblies/boxes, while others (e.g., requirements defintion, design, test related costs) can apply to all levels of hardware indenture.

Replicated, flight-quality hardware units used in simulators, test beds and the like are defined as recurring costs in the next section. However, additional engineering units used to support software development or other purposes are classified as nonrecurring.

The costs for correcting design deficiencies (redesign, rework and retest) have, in many cases, been embedded in recurring costs, along with rework and retest due to workmanship problems. However, there have been noteable cases where correction of design problems (to include manufacturing and test procedures) has required major effort and cost. For this reason, the definitions were not compromised by classifying all of these efforts as either nonrecurring or recurring. Rather, the efforts associated with design deficienies are classified as nonrecurring, while those associated with production workmanship (i.e., on protoflight, flight and spare hardware units) are classified as recurring. This approach is different from that employed by the DoD instructions for Contractor Cost Data Reports (CCDRs), where (all) redesign, rework and retest is classified as recurring.

Table 5-1 Space Hardware Nonrecurring and Recurring Cost Definitions

Nonrecurring Cost	Requirements definition, engineering design & analysis Manufacturing tooling, fixtures and procedures Development units (breadboard, engineering, qual), including parts and materials Handling and test support equipment, simulators Development, qualification and life testing Development and acceptance test procedures Engineering support for development unit fabrication, assembly & tests Refurbishment (both qual and protoflight units) Redesign, rework & retest to correct design flaws Maintenance of test & support equipment up to first launch
Recurring Cost	Hardware production units (protoflight, flight, spare) Production unit parts & materials, assembly and testing Engineering support for production unit fabrication, assembly & tests Spare parts, materials and subassemblies for production units Rework & retest due to workmanship problems Maintenance of designs and production procedures Maintenance of test and support equipment after first launch

Refurbishment of qual and protoflight units could be viewed as recurring in that it is part of the cost of producing operational units from qual and protoflight units. However, virtually all refurbishment is a “one-time” activity and is therefore defined as NR. Also, this convention can lead to a more accurate understanding of the true cost of producing flight and spare units. Specifically, if the refurbishment costs are treated as recurring and collected in the same cost accounts as true recurring costs, such as fabrication, assembly and test of flight units, then the resulting total costs would be “contaminated” and therefore difficult to use as the basis for estimating the cost of producing additional units.

Ground support equipment maintence may begin before the first spacecraft launch and is truly recurring in nature. However, the SSCAG member contractors recommended the compromise that it be treated as nonrecurring up until first launch for ease of cost accounting. Thus, this relatively small amount of maintenance effort can be viewed as part of the GSE development cost. Earlier transition dates could be established if desired, since (presumably) one or more cost accounts would need to be created to capture maintenance after the first launch. Replacement or improvement of GSE is a nonrecurring cost, regardless of when it occurs.

Space Hardware Recurring Costs

Space hardware recurring costs (highlighted in Table 5-1 above) are primarily the costs of producing protoflight, flight and spare hardware units, including:

- Manufacture, assembly and acceptance testing of operational space hardware units (protoflight, flight and spare) and flight-configured units used in ground testing and simulators
- Parts and materials used in these units
- Engineering support for manufacture and test of these units
- Spare piece parts, assemblies and complete spare units
- Incremental costs of testing protoflight units beyond that required for acceptance test
- Rework and retest, if necessary, to correct "workmanship" problems on operational units
- Maintenance of design and analysis documentation, hardware data and test procedures
- Maintenance of ground support equipment after first launch

Spacecraft components (e.g., flight computers), test equipment and simulator hardware installed at ground stations to support system operations are classified as recurring. If the test equipment and simulator hardware is primarily to support system development at the ground station, then it is classified as nonrecurring.

The cost of "setting up" prior to fabrication and testing the first hardware unit (qual, protoflight or flight) represents a dilemma. In the short-term sense, i.e., in the time span of a single acquisition contract, it is nonrecurring, although multiple ("periodic") setups can sometimes occur within a single contract. In the longer term, over multiple production contracts, setup is typically recurring, in that setups are usually needed at the beginning of each contract. Setup effort is defined here as recurring. This applies to both the physical setup as well as administrative, or "paper," setup effort.

With this convention for setup costs, recurring costs for an acquisition contract can be used as the basis for estimating all of the recurring costs for production in subsequent contracts. Of course, this is predicated on the assumption that identical setup effort(s) will be needed for the follow-on contracts. The alternative convention, defining setup as nonrecurring, would lead to difficulty in estimating future production costs (unless the setup costs were recorded separately from other hardware development costs). A refined cost accounting approach could isolate setup costs in a third "nonrecurring recurring" category, but at the expense of additional cost accounting effort.

If the same setup is used to produce a qualification unit and then to produce the first flight unit, its cost may be difficult to differentiate from other costs related to producing the qual unit. One solution would be to create a separate cost account for the setup in these cases, particularly if the setup effort is fairly large or significant. Relatively few qual units have been produced in most recent space system developments, in deference to the protoflight approach to design qualification, so this case would be infrequent.

Incremental effort associated with protoflight testing is truly nonrecurring in nature. However, SSCAG has compromised by including it in recurring cost because it is typically (a) small compared to the total acceptance level test effort and (b) difficult to capture separately from the acceptance test effort.

5.2. Ground Station and Ground Terminal Hardware

Ground station and ground terminal hardware nonrecurring and recurring costs, identified in Table 5-2, generally follow the definitions above for space hardware. Requirements definition and allocation, design, analysis and development testing are all nonrecurring costs. Also, the costs for fabrication, assembly and testing of development hardware components and units that are not a part of the operational ground station are nonrecurring.

Table 5-2 Ground Station Hardware Nonrecurring and Recurring Cost Definitions

Nonrecurring Cost	Requirements definition, engineering design & analysis Development unit parts, materials, fabrication, assembly and tests Engineering support for development unit fabrication, assembly & tests Redesign, rework & retest of development hardware due to design flaws Simulators and test equipment specifically for supporting development
Recurring Cost	Parts, materials, fabrication, assembly, installation and tests of all installed operational hardware Engineering support for fabrication, assembly, installation & test of operational hardware Spare parts, materials and subassemblies and assemblies Rework, reinstallation and retest of operational hardware due to workmanship problems Simulators and test equipment specifically for supporting system operation, or for supporting both development and operation

Just as protoflight units are treated as recurring for space hardware, all installed (operational) ground station hardware components are treated as recurring. This applies to single ground stations as well as multiple, replicated ground stations. In addition, all spare parts, materials and spare integrated hardware units are recurring. This applies to “initial” spares acquired or produced early in the life cycle as well as “replacement” spares acquired or produced later in the system operational phase. These definitions support the primary objective of enabling recurring costs for replicating ground stations to be separated from ground station development costs.

Simulator and test equipment used in ground stations is also considered recurring unless its primary purpose is to support system development. If only one unit or set of equipment is produced and it is used for both development and subsequent operations and support, then it is recurring. However, if multiple, identical units are produced to support both development and O&S, then the first unit is nonrecurring and the rest are recurring.

These definitions apply to both initial ground station development and subsequent improvements, upgrades and equipment replacements (often referred to as "recapitalizations", or "recaps"). Maintenance of ground station hardware is addressed in Section 5.5.

The primary motivation for capturing the costs of all installed hardware (plus spares, simulator hardware and test hardware used to support system O&S) as recurring is to provide a basis for estimating the costs of the same, or similar, hardware elements within replicated ground stations or terminals. Another motivation is to provide an understanding of how much of the total cost for the ground station/terminal hardware represents development effort, as opposed to producing the operational system components and subsystems.

It should be noted that segregation of ground hardware efforts into NR and Rec components is likely to be difficult at low levels of indenture (board and "drawer" levels). Therefore, somewhat less precision, or accuracy, in NR/Rec cost segregation for ground hardware, as opposed to space hardware, might be anticipated.

5.3. Software Costs

The following definitions apply to ground station software, spacecraft flight software, spacecraft and ground segment test software and simulator software. The definitions are summarized in Table 5-3 on the next page. Software nonrecurring costs include all efforts required to design, develop and test qualified software, including:

- Requirements definition, top level design, algorithm development and analysis
- Design, coding and unit test of software modules and components
- Licenses for COTS software that is integrated into CSCs/CSCIs for development purposes
- Licenses for COTS software that is used in development facilities or to support development at ground stations
- First-time integration and testing of CSCs and CSCIs
- First-time integration and test of CSCIs at subsystem, segment and system levels of integration
- Requirements verification, including independent validation and verification (IV&V)
- Correction of software defects and software enhancements until the system-level Initial Operational Configuration (IOC) is achieved
- Development of test software and other software for supporting development efforts
- Design, analysis, development, test and other software documentation
- Software configuration management and quality assurance
- Maintenance of spacecraft development and production support software until first spacecraft launch

As in the hardware case, these development efforts are intended to span all levels of software and hardware-software integration. Thus, support by software developers for integrated hardware-software development testing, up through the total system level, is nonrecurring.

System-level IOC is typically achieved when all elements of the system have been fully integrated and successfully tested. Thus, all portions of the system and data streams that may have been simulated in earlier phases are replaced by the "real" system elements and data streams as a condition of achieving IOC. There may be a period of on-orbit spacecraft checkout after IOC, followed by normal system operations.

Table 5-3 Flight and Ground Station Software Nonrecurring and Recurring Cost Definitions

Nonrecurring Cost	Development of delivered operational S/W Software test procedures Software configuration management, quality assurance & documentation Support for/conduct of first-time software I&T with hardware and subsequent higher level I&T Test, development support and spacecraft production support S/W: development and maintenance up to first launch Commercial S/W licenses for development facilities and ground station development support Correcting deficiencies and developing enhancements in ground station And flight software after installation up to IOC
Recurring Cost	Installation and check-out of developed S/W in ground station If first-time S/W I&T is performed at ground station, then it is NR Maintenance of S/C production & test support S/W after first launch Acquisition of stand-alone and bundled COTS S/W in operational system Maintenance of flight S/W (usually embedded in acquisition contracts)

Maintenance of software that is related to, or supports, production and testing of operational spacecraft is nonrecurring before the first spacecraft launch and recurring after that. This compromise matches that applied to spacecraft ground support equipment in the previous section. Maintenance of spacecraft flight software and ground station software is addressed in Section 5.5, Operations and Support Costs.

Costs for stand-alone COTS software installed in operational ground stations and software purchased bundled with hardware are both recurring costs – except for COTS software installed and used exclusively to support development as opposed to operation and maintenance.

Costs for installation and testing of previously-developed software at operational ground stations are recurring costs. The software may include both custom software and COTS software. However, if first-time integration of the software occurs at the ground station, as opposed to a separate development facility, then the installation and testing is nonrecurring. This definition clearly leaves a "gap" in the basis for estimating replicated ground stations. If the cost of the (nonrecurring) first-time integration and testing is recorded or estimated separately from the rest of the software nonrecurring (development) costs, it can be used as the basis for estimating I&T costs for replicated ground stations/terminals.

As stated above, costs for correcting software deficiencies and enhancing the software up until System IOC are nonrecurring. Since installation of previously-developed software at ground stations is recurring, the nonrecurring costs for deficiency correction and enhancements may occur later in time (than the installation costs). This definition applies to both the first ground station and to subsequently installed ground stations, each with its own IOC. Specifically, costs for correction of deficiencies and for enhancements required by, or associated with, a subsequent ground station are nonrecurring until the subsequent IOC is achieved.

5.4. SEIT/PM Costs

This section addresses system engineering; integration and test; and program management for space systems. As indicated at the beginning of Section 5, SEIT/PM generally occurs at multiple levels of indenture, including the overall system, system segments, and subsystems or functional entities within segments. The definitions are intended to address all of these SEIT/PM levels, although some portions may not apply to all levels.

It should be noted that some of the tasks identified under SE, I&T and PM may actually be located within one of the other areas (SE, I&T or PM) of a given Contract Work Breakdown Structure. For example, quality assurance is addressed in these definitions as a program management cost, although it is sometimes booked by contractors under system engineering (e.g., as part of “product assurance”). Test planning is addressed under Integration and Test below, although it may be viewed as a system engineering activity at high levels. Configuration management is addressed under system engineering, although it could be a PM effort. These definitions are not intended to provide guidance on where specific SEIT/PM efforts should be located within a contract work breakdown structure.

Many SE and PM tasks, such as technical direction and business/financial management, may be “level of effort” type activities, with no inherent criteria for identifying nonrecurring vs. recurring effort. Arbitrary termination of nonrecurring and initiation of recurring cost accounting at a specific time, such as the system-level CDR completion date, is the usual solution to this problem.

System Engineering

System Engineering (SE) nonrecurring efforts, highlighted in Table 5-4 on the next page, include requirements definition, allocation, flow-down and verification; system concept and configuration definition; specification and interface control document (ICD) development; top level design trade-offs; analyses to predict system performance, reliability, availability, maintainability, survivability and effectiveness; development of verification plans and procedures; and operational concept and training plan development.

Mass properties, electrical power, communication link power, technical performance, and reliability/availability budgets are usually developed by system engineering organizations as nonrecurring activities. Other types of plans, analyses and budgets may also be produced by system engineering organizations as nonrecurring activities, depending on customer requirements, product type and corporate policies. Examples are development, manufacturing, verification, training and security plans; processing time and memory budgets; and security and launch range safety analyses. Costs for all of these system engineering products are nonrecurring. Establishing parts, materials and processes infrastructure is also nonrecurring.

The development and maintenance of specifications and interface control documents is a nonrecurring effort until they are completed – nominally when all "to be defined" (TBD) and "to be reviewed" (TBR) portions are removed. Development and maintenance of plans, budgets and other system engineering products is nonrecurring until design completion. Design completion is typically after Critical Design Review (CDR), because detailed design is usually not complete at CDR.

Table 5-4 System Engineering Nonrecurring and Recurring Cost Definitions

Nonrecurring Cost	Requirements definition, allocation and flow-down Specifications and ICDs – initial development & maintenance until all TBDs and TBRs are removed Top level design tradeoffs Analyses and simulations of system performance and effectiveness, reliability, availability, maintainability, survivability Operations concept development System configuration management during development Mass, power, link margin & reliability budgets -- initial development & maintenance until design is complete Processing time & memory budgets Development, production, verification, training and security plans Parts, materials and processes – establishing procedures, etc. Requirements verification through design completion, including demonstrations and IV&V Technical change management (RFIs, ECPs, etc.) -- NR tasks supporting changes, handling nonrecurring-only changes Functional and physical configuration audit (FCA/PCA) support
Recurring Cost	Maintenance of specs, ICDs, plans & budgets -- after all TBDs and TBRs are removed Performance, S/W maintenance productivity metrics Analysis updates System configuration management after design completion Mass, power, link margin & reliability budgets – supporting production Support for production testing, rehearsals and OT&E Requirements verification after design completion Launch readiness reviews, support for product acceptance Technical change management (RFIs, ECPs, etc.) -- supporting recurring portions of changes and recurring-only changes

Management of technical changes and engineering change proposals (ECPs) which are nonrecurring in nature – or the portions which are nonrecurring – is a nonrecurring SEIT/PM effort.

Recurring system engineering efforts include maintenance of specifications, plans, budgets, and ICDs; analysis updates to reflect the current system configuration; configuration management after design completion; maintenance of technical performance metrics (TPMs) and software productivity metrics; test monitoring and verification of requirements satisfaction; ECP management for recurring activities; conduct of launch readiness reviews; and support for operational test and evaluation activities. Some of these activities can take place over the entire operational phase of the system life cycle.

Integration and Test

Integration and Test nonrecurring efforts identified in Table 5-5 include development, qualification and life testing, and development of plans and procedures for both development tests and acceptance tests. Development and production of ground support equipment, test equipment, test software and simulators (except as noted above) is also nonrecurring. [Note: Other names for GSE are Aerospace Ground Equipment (AGE), Mechanical Aerospace Ground Equipment (MAGE), Electrical Aerospace Ground Equipment (EAGE) and Special Test Equipment (STE)].

Table 5-5 Integration and Test Nonrecurring and Recurring Cost Definitions

Nonrecurring Cost	Development, qualification and life testing – all levels and system segments Development and acceptance test plans and procedures Ground support and test equipment, test software, simulators I&T for correcting software problems prior to IOC
Recurring Cost	Spacecraft acceptance testing Integration and test of previously-developed software – in ground stations/terminals and follow-on spacecraft

Spacecraft recurring I&T efforts are primarily those for acceptance testing. However, incremental effort for protoflight testing, above and beyond the effort required for acceptance testing, is also recurring. Software I&T efforts are nonrecurring except for I&T of previously-developed software in operational ground sites and in follow-on spacecraft, which are recurring. However, I&T effort prior to system IOC that is devoted to, or associated with, correcting design deficiencies is nonrecurring.

Program Management

Program management (PM) nonrecurring and recurring costs are identified in Table 5-6.

Table 5-6 Program Management Nonrecurring and Recurring Cost Definitions

Nonrecurring Cost	Management activities applied to/associated with NR work – technical direction, financial mgmt, subcontract mgmt, quality assurance, computer support Procedures and infrastructure – Cost accounting, data mgmt, parts & material acquisition, security, safety, facilities
Recurring Cost	Management activities applied to/associated with NR efforts

Nonrecurring PM costs include those for technical direction, business/financial management, subcontract management, quality assurance, computer support and other support functions as they apply to, or are associated with, all program nonrecurring effort/tasks. They also include costs for establishing procedures and infrastructure for cost accounting, data management, acquisition of parts and materials, program security, safety and facilities. Recurring PM costs include those for technical direction, business/financial management, subcontract management, quality assurance, computer support and other support functions as they apply to recurring effort/tasks.

5.5. Operations and Support Costs

O&S nonrecurring and recurring costs are identified in Table 5-7. Nonrecurring O&S costs include development of operations and maintenance concepts; procedures; and training materials, aids and simulators. Integrated logistics support (ILS) analyses data are also nonrecurring. In addition to these activities and costs that take place during initial system development, the same types of activities and costs incurred as part of major redesigns, incremental capability increases, recaptitalization and commercial software upgrades/replacements are also nonrecurring.

Table 5-7 Operations and Support Nonrecurring and Recurring Cost Definitions

Nonrecurring Cost	Concept of operations and maintenance development Operations & maintenance procedure development Development of training materials, simulators and aids Initial ILS analysis and data Upgrades/changes to above associated with: Major redesigns, capability increments Recapitalization, operating system changes
Recurring Cost	System operation costs – operations, training, system engineering & management staff Hardware maintenance & recapitalization Software maintenance – including COTS license costs

	Software modifications/fixes after IOC Reproduction and updating of technical data and manuals Procurement/production of hardware spares Rehearsals and initialization of spacecraft

All costs for operating and maintaining the system are recurring. These include:

- Staffing ground stations with operations, maintenance and support (infrastructure) personnel
- Training, system engineering and management efforts supporting operation and maintenance
- Maintenance of operational, testing and simulator hardware in ground stations and terminals
- Maintenance of spacecraft flight software and ground station/terminal software – including COTS license costs
- Vendor maintenance agreements (VMAs) and recapitalization (hardware and software)
- Production or purchase of spare ground station hardware
- Reproduction and updating of technical manuals and data
- Launch rehearsals and post-launch initialization of spacecraft on orbit

Spacecraft and ground station/terminal software recurring maintenance costs include error/deficiency correction and minor improvements/enhancements/upgrades after system IOC. Correction of errors/deficiencies before IOC is nonrecurring, while other maintenance activities (e.g., minor improvements) before system IOC are recurring.

Flight software is usually maintained under spacecraft production contracts, as opposed to separate O&S CLINs or contracts.

5.6. Skill Retention and Contract Closeout Costs

Costs for retaining critical skills needed for future spacecraft production contracts are usually nonrecurring in nature. Typical activities include minor design improvements, replacement of obsolete parts and development of new capabilities for the next spacecraft production lot. Skill retention costs are often recorded under dedicated cost accounts as opposed to being recorded against hardware, software or SEIT/PM accounts. In some cases, separate contract CLINS or even separate contracts may be used to capture these costs. In some cases, support for current production, such as problem resolution, may be recorded on skill retention CLINs or contracts. In these cases, the rules for redesign, rework and retest of spacecraft hardware should be applied. Thus, recurring costs related to workmanship problems may be recorded under skill retention.

Contract closeout costs may have both nonrecurring and recurring elements. For contracts where a follow-on production contract is planned, the costs may be primarily or exclusively recurring in nature. However, closeout costs for the final production contract of a space system/program may have substantial nonrecurring costs. Likewise, contracts which are terminated for the customer’s convenience would likely have significant amounts of nonrecurring closeout costs. In many cases, separate cost accounts are used to keep closeout costs separate from other costs.

6. Guidelines for Nonrecurring and Recurring Cost Accounting

This section addresses typical situations and problems encountered in recording nonrecurring and recurring costs. Both NR/Rec cost segregation during the execution of acquisition contracts and ex post facto segregation after contract completion are discussed.

Section 6.1 provides guidelines for setting up contract cost accounts and lower-level cost collection elements - both initially in the proposal phase and subsequently during contract execution. Section 6.2 addresses segregation of subcontracted efforts and Section 6.3 addresses parts and material cost segregation.

Section 6.4 addresses implementation of cost accounting practices during contract execution. Section 6.5 provides guidelines for estimating the amount of NR and Rec costs in accounts which contain a mixture of both NR and Rec costs. This is typically done late in the production phase or after contract completion by cost analysts.

6.1. Setting Up Cost Accounts

This section provides guidelines for setting up initial cost account structures for a new contract. Cost accounts are typically set up in the proposal phase according to customer guidance in Requests For Proposal (RFPs). The guidelines are also applicable to new cost accounts created throughout the contract execution phase. Substantive contract changes, usually preceded by an Engineering Change Proposal (ECP), often entail both nonrecurring and recurring efforts. Sometimes additional accounts are created when serious development or production problems arise. Both nonrecurring and recurring efforts may be embedded in problem resolution.

The conventional approach to recording NR and Rec costs is straightforward: creation of separate NR and Rec job numbers for each work area where cost accounting is desired. For example, separate NR and Rec J/Ns might be created for the engineering effort related to an individual hardware assembly/box or the design effort for a flight software CSCI. Similarly, pairs of separate NR and Rec J/Ns might be created for management, system engineering and integration and test at the subsystem, spacecraft, segment and system/contract levels of indenture.

As indicated in Section 3, actual costs are collected into cost accounts for purposes of analysis and reporting under Earned Value Management Systems (EVMS). Earned value is measured at the cost account level. It is beneficial from a manager's perspective to have separate NR and Rec CAs. This allows evaluation of cost and schedule variance for both development and production efforts. While this could cause the number of CAs to double, the benefit could outweigh the marginal increase in administrative costs.

If NR and Rec costs are not segregated at the CA level, the next preferred method is to segregate them at the job number, work order or cost charge number. This method allows NR and Rec costs to be fully recorded and reported, even though earned value analysis is not performed for each. Thus, when NR and Rec J/Ns are used, historic data collected will readily support development of cost estimating relationships (CERs), estimating by analogy and other uses depending on NR/Rec segregation.

From the cost analyst's viewpoint, cost accounts should be established across the contract to capture NR and Rec costs separately for all significant end-items and SEIT/PM efforts. However, this goal may be at odds with the program manager's goal to arrange cost accounts to support management of the work to be performed on the contract. This often takes the form of accounting for multiple end-items within a set of "common" cost accounts, as opposed to establishing separate cost accounts for each individual end-item, because the end-items are managed by a single person or work center.

In some cases, intermediate level NR/Rec cost segregation may not be required by the customer, but the contractor may wish to define lower-level accounting elements to achieve NR/Rec segregation for cost estimating purposes. This is particularly true in cases where a follow-on contract is anticipated and/or when the actual costs might be used by the contractor as the basis for cost proposals for new systems.

The total number of cost accounts and the number of separate CWBS cost elements for which cost data is required by customers directly impact the resources required for contract cost accounting. If the contractor has a compelling need to reduce the number of cost accounts by eliminating separate cost accounts for NR and Rec costs, then a natural choice would be to eliminate them for management and system engineering – while retaining NR/Rec segregation by cost account or hardware and software end-items (at least most of them) as well as I&T. The reasons for this approach are: (a) SE and PM costs are the most difficult types of cost accounts to segregate and (b) the value of segregating them (to both contractor and government) is typically much less than the value of segregating end-item and I&T costs.

Another "trade-off" with regard to reducing the total number of cost accounts may pit the need for NR/Rec segregation against the need for capturing true end-item costs. For example, after eliminating separate NR and Rec cost accounts for SE and PM (at multiple CWBS levels), the next step in reducing the number of cost accounts might be to eliminate NR/Rec segregation for end-items. An alternate step would be to retain NR/Rec segregation for end-items, but aggregate them into higher-level groupings.

An example of this trade-off is (1) grouping all power control and conditioning electronics into two (NR and Rec) cost accounts, or (2) assigning each individual box its own NR+Rec cost account. From the cost analyst's viewpoint, Option 2 would be preferable, with approximate NR/Rec segregation performed ex post facto as described in Section 6.5. However, fewer cost accounts are eliminated in this option, assuming the average end-item grouping has more than two items in it. Option 1 might be more desirable if estimating is routinely at the electronics group level, as opposed to the individual box level. As indicated above, Option 1 might also be preferred by program managers to align cost accounts with the lowest level management responsibility areas.

6.2. Subcontracts and Interdivisional Work

The SSCAG survey revealed that separate NR and Rec cost reporting by subcontractors to prime contractors is usually not practiced on a regular basis, but rather "occasionally". This, of course, can leave a major gap in the prime contractor's knowledge of the bottom line NR and Rec costs for an acquisition contract, particularly when major payload elements are subcontracted.

Ideally, NR/Rec cost segregation for interdivisional work performed by the prime contractor should be treated the same way as work performed within the contracting division. However, differences in accounting systems and management practices may result in less than uniform segregation. If the interdivisional work is relatively large, it represents the same potential for missing knowledge as do subcontracts.

Customer “flow-down” requirements for NR/Rec cost segregation are the first, and most important, step in obtaining realistic cost segregation on major subcontracts and interdivisional work. If this is not done, the prime contractor can take appropriate steps to voluntarily obtain some level of knowledge of NR and Rec cost magnitudes. One approach is to require some form of segregated NR/Rec cost reporting, even if informal, for all subcontracts and interdivisional efforts, or at least those above a given cost threshold.

The prime contractor could apply its own NR/Rec split guidelines based on time to those subcontracts where no NR/Rec segregation is obtained from the subcontractors. However, care must be taken when some of the subcontracted hardware production, for existing hardware end-items, begins long before CDR.

Since the primary purpose of segregating NR and Rec costs is to support future cost estimates and proposals, prime contractor estimating organizations are well advised to vigorously pursue the most uniform cost segregation practical across the prime contract, major subcontracts and significant interdivisional effort – even if customer requirements do not specifically require uniformity.

6.3. Parts and Materials

The SSCAG survey indicated that separate NR and Rec cost accounting for parts and materials (P&M) is not as prevalent as that for in-house labor. When practiced, separate P&M cost accounting is more likely to be at intermediate to high levels of indenture, as opposed to detailed (e.g., individual part types or orders).

Some form of NR/Rec cost segregation for P&M via cost accounting is recommended, particularly where a substantial amount of development hardware (primarily engineering units and occasional traditional qualification units) is produced.

Even though parts and materials costs can only be allocated into NR and Rec components accurately at the lowest level of detail, approximate methods at aggregate levels involving judgment may be sufficiently accurate to justify the associated effort. For example, costs for parts used in both development and production units could be recorded in separate accounts from those needed only for production units (protoflight, flight and spares). Another approach is to create NR and Rec cost accounts for the most expensive parts, and applying approximate segregation to the remaining parts.

Some companies routinely keep separate cost accounts for development and production parts. This makes sense particularly when lower grade parts are used in engineering units than in production units.

6.4. Implementation

The keys to good NR and Rec cost accounting are first setting up NR and Rec accounts to properly segregate the bulk of the contract costs, developing clear guidelines for "charging" those accounts and following the guidelines in a disciplined manner. The guidelines should be as simple and unambiguous as possible so that individual workers and managers have a common understanding of how to record their work hours. The following are examples of guidelines for recording hours properly:

- Charge NR accounts when establishing requirements, designing and performing engineering analyses
- Charge NR accounts when working on, or supporting work on, development (engineering, qual) units
- Charge Rec accounts when working on, or supporting work on, production units (protoflight, flight and spares)
- Charge NR accounts when designing, coding and unit testing software – and when supporting first-time integration and test activities at all levels of indenture

Simpler but less accurate guidelines can be based on time (see Section 6.5 below) rather than type of activity. However, this practice is not recommended because it subverts the very objective for which separate NR and Rec accounts are set up in the first place – to capture "true" NR and Rec efforts – whenever they may take place.

Given guidelines based on the type of activity as exemplified above, NR cost accounts should be left "open" after CDR to capture completion of design effort and other NR efforts that occur after CDR. Similarly, Rec accounts should be open as soon as work on production units begins – even if that occurs prior to CDR.

Considering typical management pressure to close out cost accounts, when all known/planned design and analysis is complete and hardware production is underway, it is reasonable to close out most NR accounts. However, some open NR accounts should be available to capture NR efforts during later production phases, such as redesign (plus associated rework and retest). Thus, proper cost accounting guidelines will include the criteria for opening and closing NR and Rec accounts.

6.5. Cost Accounts Containing Both Nonrecurring and Recurring Costs

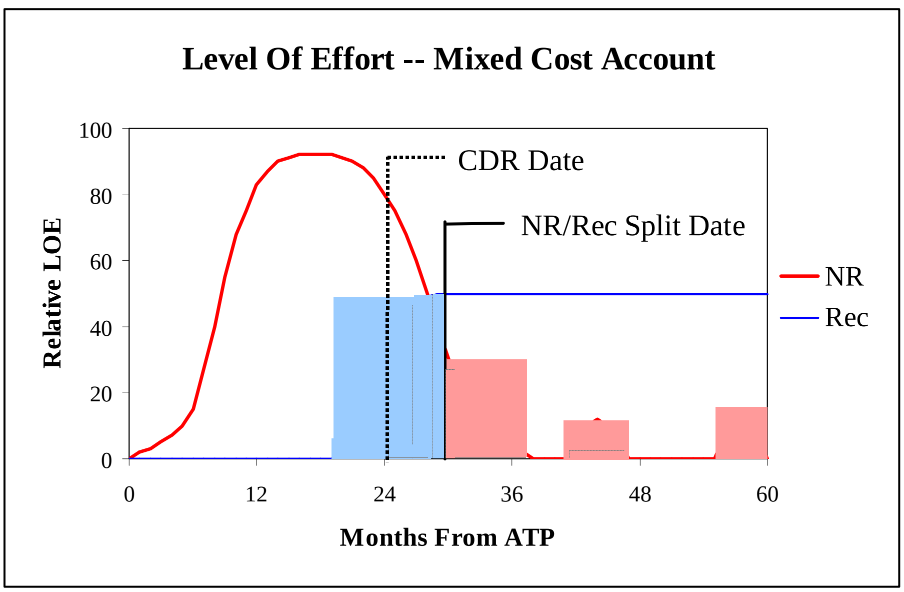
When single accounts are used to record both NR and Rec actual costs, the problem of segregating these costs into NR and Rec components "after the fact" arises. The most common method of approximating the NR and Rec components of the recorded costs has been to treat those costs prior to CDR as NR and those after CDR as Rec. This approach is most frequently taken for SEIT/PM level-of-effort tasks (see Section 5.4 above). However, it has also been applied extensively to hardware end-item development and production.

When this approach is used, presumably the specific CDR most applicable to each account is selected. For example, the CDR date for a spacecraft electrical power subsystem (EPS) would be naturally be used to segregate NR and Rec costs for those "mixed" accounts applicable to the EPS (as a subsystem) and/or its components. Higher-level CDR dates (spacecraft bus, space segment) might result in less reliable NR/Rec cost segregation for EPS accounts.

The SSCAG survey suggested that the use of CDR dates to segregate the NR and Rec cost components of mixed accounts tends to underestimate NR costs and overestimate Rec costs in general – even when some recurring production work takes place before CDR. This is primarily because (1) not all detailed design may be complete at CDR; (2) engineering unit manufacture and testing, and qualification testing may not be complete; and (3) latent design deficiencies requiring correction typically occur after CDR. Since protoflight level testing is not performed until after the protoflight unit fabrication and assembly are completed, nonrecurring effort to correct latent design deficiencies may occur long after CDR.

For large mixed accounts, examination of the tasks performed in the periods just before and after CDR may be useful in establishing alternate dates for separating NR and Rec costs. Figure 6-5 shows hypothetical distributions of nonrecurring effort (in red) and recurring effort (in blue) over time, starting with ATP and continuing well into the production phase. This might be typical of space hardware assembly or "box"-level hardware development and production. However, higher-level efforts, including SEIT/PM, might have similar distributions.

Figure 6-5 Mixed Cost Account Effort vs. Time



As the SSCAG survey suggested, the amount of nonrecurring effort after CDR, the area under the red curve to the right of the CDR date, is significantly larger than the recurring effort prior to CDR (the area under the blue curve to the prior to CDR.

A better "dividing point" in time is after CDR, identified in the figure as the "NR/Rec Split Date". The amount of NR effort after this date, indicated by red shading, is roughly equal to the amount of

Rec effort prior to the “split date” (blue shading). Thus, the split date can be viewed as the ideal point in time with regard to obtaining realistic NR and Rec estimates from a mixed NR/Rec account.

Analyzing the activities that took place in the periods before and after CDR to establish realistic split dates is probably not feasible in most cases, particularly for large contracts with many mixed accounts. However, analysis of a few typical cases may lead to general guidelines that can be applied to all mixed accounts. For example, detailed analyses might indicate that the ideal split date is usually between three and six months after CDR.

An easier (and probably more reliable) way to develop guidelines would be to analyze typical cases where both NR and Rec accounts were established at ATP and used to record effort through first unit delivery. Comparing cost vs. time in corresponding NR and Rec accounts to establish split dates would yield fairly accurate results.

Some contractors have used post-CDR milestones such as manufacturing readiness reviews and last (planned) drawing release as NR/Rec split dates.

One typical case that should be treated carefully is a “rebuild”, where the design is intact (existing) from a previous contract and hardware production begins relatively early in the development phase. These cases will typically have relatively little nonrecurring effort and the split date approach will be unreliable.

The NR/Rec Split Date approach can also be applied to software development and maintenance efforts within the same cost account. In this case, the split date would be much closer to software product acceptance than any CDR date. Consideration should be given to cases where development of some of the software components (CSCs) is completed long before other CSCs in the same account.

7. Summary

The definitions and guidelines provided herein are intended to strike a reasonable compromise between the most rigorous possible approaches and the practical limitations typically encountered in actual space system acquisition contracts. SSCAG offers them as a starting point for government agencies and contractors to establish regularized practices regarding one of the most basic problems in space system cost estimating. More refined definitions and additional guidelines are expected to evolve as these definitions are applied by SSCAG member organizations and within the space system estimating community at large.

APPENDIX J

Establishing a Parametric Implementation Team

This appendix describes the basic principles involved in preparing and negotiating a proposal based on parametric estimating techniques, and focuses on the role the parametric implementation team plays in this process. This appendix also:

- Provides guidance on setting up a successful parametric implementation team;
- Discusses key processes related to the Government evaluation and negotiation of a parametrically based proposal;
- Explains the roles of team members;
- Lists applicable Parametric Estimating Reinvention Laboratory best practices.

Proposals and Parametrics

The processes used for developing, evaluating, and negotiating proposals based on parametric techniques are not significantly different than those used for proposals based on traditional estimating methods. The Parametric Estimating Reinvention Laboratory demonstrated, though, that those processes do require:

- Up-front preparation so all parties understand the parametric process;
- Teamwork in the form of a joint implementation team;
- Training to ensure proper use of the parametric estimating techniques being used.

The Implementation Team and the Parametric Proposal Process

What steps in the proposal process are most affected by a contractor's use of parametric estimating techniques as a basis of estimate (BOE)? Actually, the process as such does not change dramatically when a contractor uses parametric estimating techniques. The most important step is the contractor's initial decision to develop the proposal parametrically, which then leads to the establishment of an implementation team, the definition of its charter, and the determination of the buying office's role in it, all of which affect how the proposal will be evaluated and negotiated. Throughout the process, the contractor should maintain regular communications with the buying office and knowledgeable representatives from the DCMA and the DCAA, in order to ensure that the proposed estimating methodology, and the presentation of its results in the proposal, meets the customer's needs.

Contractor's Decision to Use Parametric Estimating Techniques

The first step in the parametric proposal evaluation process is the contractor's decision to use parametric techniques as the BOE. A major factor in this decision is the contractor's determination that it is feasible to develop a parametric estimating system capable of producing reliable estimates in a consistent manner. In order to develop this capability, contractors should have, or establish, a formal data collection program. The Parametric Estimating Reinvention Laboratory demonstrated that it is very important for a contractor to have an adequate, relevant database since this forms the foundation from which to predict the cost of a future project or item.

A contractor should have enough expected future business to justify the cost and effort required to implement a valid parametric estimating capability. In addition to using a parametric technique as a proposal-bidding tool, a company should consider any derivative benefits it may have, such as its application to risk analysis and target costing. Chapter 8, Other Parametric Applications, discusses these uses.

Establishing an Implementation Team

Process change or improvement is not possible without the support of the process owners and the customers of the process outputs. The Parametric Estimating Reinvention Laboratory demonstrated that the use of a joint implementation team to implement and employ parametric estimating techniques is a best practice. The team's composition generally consists of representatives from various functional departments of the contractor, buying office, DCMA, and DCAA. For example, many Parametric Estimating Reinvention Laboratory sites chartered their implementation team as a formal IPT. Regardless of whether the team is an IPT or an ad-hoc committee, it should meet and establish its objectives early in the evaluation process, before significant resources are expended. Early organization and regular meetings ensure a clear focus for the company, and the opportunity for the primary Government customers to express concerns and expectations.

The company and its customers may form the team in different ways. One approach is to form an executive-level group made up of senior management personnel from the contractor, customer (including program management personnel), DCMA, and DCAA. This group, often referred to as a management council, provides overall policy direction, resources, training authorization, and high-level organizational support. The executive-level group selects staff personnel to form a working-level implementation team. Another method involves the establishment of a formal IPT which develops the parametric estimating system, sets up joint training for all team members, recommends estimating system policy changes, and develops and evaluates the proposals.

The most important feature of the team is not how it is formed, but that its members are dedicated, open-minded, and interested in committing the time and energy to investigate the opportunity of developing a parametric estimating capability. This increases the likelihood that the customer will accept a parametrically based proposal as a valid basis for negotiating a fair and reasonable price.

Responsibilities of the Implementation Team

Once established, the implementation team must accomplish a number of tasks, including:

- Obtain approval from the contractor and Government management to establish the required resources. Prior to a full commitment of resources, contractors typically perform a return on investment (ROI) analysis to illustrate the costs and benefits of implementing a new estimating technique. If the analysis justifies further pursuit, the company seeks internal management commitment.
- Develop or obtain training for team members (and others as needed). Training can include an overview of parametric estimating techniques, as well as detailed training on the specific CER or model being implemented.
- Develop a methodology for calibrating and validating the parametric model. This requires either developing expertise or obtaining it from other sources.
- Establish rules for joint proposal development and proposal negotiation.
- Establish approved parametric estimating policy and procedures.
- Address regulatory issues, such as TINA, FAR, and CAS.
- Resolve as many differences as possible between the Government and contractor. However, some issues may remain that must be settled by negotiation.

Establishing an Approved Parametric Estimating System

Prior to using parametric techniques on proposals, contractors should establish effective estimating system policies and procedures that comply with Government procurement regulations. Chapter 7, Government Compliance, discusses criteria for establishing adequate policies and procedures.

Proposal Evaluation

Based on FAR 15.402(a), contracting officers are responsible for determining and negotiating reasonable prices, with input and assistance from DCMA, DCAA, and others. The contractor's task is to provide information and supporting data for this determination. When a parametrically-based proposal is received, the contracting officer should determine if it is suitable as a basis for negotiation. The contracting office may not have the necessary knowledge to make this determination due to limited training or knowledge of parametric estimating. If this is the case, then the contracting officer's response should not be to declare the proposal unsatisfactory, but instead to find the personnel or information needed to make an appropriate determination. For example, DCAA provides financial advisory services to contracting officers concerning such issues as the adequacy of proposal submissions based on parametric techniques, and the validity of a contractor's parametric estimating system.

Contractors can use parametric techniques to generate cost breakdowns in varying formats. The solicitation clause at FAR 52.215-20, "Requirements for Cost or Pricing Data or Information Other than Cost or Pricing Data," permits contracting officers to tailor proposal formats as needed. A contractor's proposal submission should provide details on the parametric technique used, model inputs, database, and statistical information sufficient to provide a basis for evaluating the technique's merits. Again, contractors can facilitate this process by ensuring that the customer, DCMA, and DCAA are part of the implementation team. Chapter 7, Government Compliance, discusses alternative proposal formats.

The offeror must support its proposal, which includes educating the customer on its practices and procedures. However, the customer also has a responsibility to maintain its knowledge of the techniques and technologies being used within Industry. If the proposal is large enough, an on-site fact-finding session may be needed to fill any gaps in the proposal supporting data. If data are not sufficient to support an evaluation, the contracting officer may return the proposal to the offeror. After a valid proposal is received, the contracting officer will typically assemble a negotiation team. This could consist of an appropriate mix of functions, such as price analysts, auditors, financial and technical advisors, and other parametric experts.

The negotiation team performs a number of tasks related to proposal parametrics including:

- Evaluating the calibration and validation of parametric tools;
- Assessing the accuracy of parametric inputs and their appropriateness as cost drivers;
- Checking the validity of the parametric database as a BOE;
- Evaluating the quality, accuracy, and relevancy of the data in the database;
- Examining the nature and reliability of the contractor's parametric estimating system

Negotiating a Parametrically-Based Proposal

If the procurement is sole-source, then the proposal is subject to negotiation regardless of the estimating method employed. Indeed, any acquisition situation, other than sealed bidding, contains some elements of negotiation. Neither parametric estimating, nor any other traditional estimating technique, establishes the one and only price. Regardless of the estimating method used, the goal is to produce an overall estimate that is fair and reasonable for the supplies and services being acquired. If the implementation team has done its work well, there should be no issues of fact regarding the use of a parametric estimating technique. The only questions should concern decisions about its use and application. For example, a properly calibrated and validated model should generally be acceptable to the contracting officer as a BOE if properly documented. However, the model inputs (e.g., weight, size, complexity) would be subject to validation by advisors to the contracting officer, and to negotiation. Other aspects of the proposed price such as profit, economic escalation, adjustments for future changes, and decrement factors for proposed purchased items are subject to negotiation as well.

The Procuring Contracting Officer (PCO) should conduct negotiations or specifically delegate pieces of the negotiation to the Administrative Contracting Officer (ACO). A parametric model should never be presented as a "black box" that produces the one and only price the buying office must accept without review and evaluation. In the end, the point of any negotiation is to settle on a reasonable price, not to come to agreement on a model, data input, or cost element.

The ACO/PCO should assemble a team including price analysts, DCMA, and DCAA to evaluate the proposal and support negotiations, and Figure J.1 provides a suggested list of actions the team should take when addressing a parametrically based proposal.

1. ***Evaluate*** procedures and results of calibrating and validating the model or CERs used to build the proposal.
2. ***Verify*** data collected and used in the model.
3. ***Determine*** if the model produces estimates that are logical, consistent with history, and are properly supported.
4. ***Obtain*** inputs on costs not estimated by the model.
5. ***Compile*** all inputs and develop a negotiation position.

Figure J.1 Areas to be addressed by the Contracting Officer's IPT

Chapter 7, Government Compliance, provide additional information on Government evaluation procedures.

Competitive Proposals and Documentation

Competitive Proposals

Parametric estimates may also be used to support competitive proposals. A key point in providing a competitive proposal is to show that the proposed price includes all aspects of the required work, and that it is adequate to complete the contract (this is known as cost-realism). Therefore, the offeror must show that the resulting estimate is realistic, and addresses all aspects of the required effort. A parametric model can do this, provided it is properly calibrated and validated. Even though the level of detail to support a realism determination may not be as great as a sole-source proposal, the offeror should be able, and willing, to show that the model is valid, inputs are appropriate and correctly estimated, and results are realistic. Chapter 8, Other Parametric Applications, discusses cost-realism.

Documentation

The Government documentation requirements do not change with a contractor's use of parametric estimating methods. A Government buying office must still produce a Price Negotiation Memorandum (PNM). The only real difference is in the data used to support the negotiation positions. All parametric tools and their major inputs should be addressed at a level of detail appropriate for the value and complexity of the specific procurement. Documentation must show and support how proposed and objective prices were developed, and that the negotiated price is reasonable. The fact that the model produces a price estimate does not establish the reasonableness of the model's output.

The Implementation Team After the Proposal Process

After a parametric tool has been developed and accepted, or a proposal using it negotiated, the implementation team may either disband or become an informal group. It is recommended that the team remain in some form (administered by the local DCMA and DCAA) and monitor and periodically reevaluate the tool, in order to maintain its quality and usefulness in future proposals and negotiations. The monitoring process is similar to that used with FPRAs (discussed in Chapter 7, Government Compliance).

Lessons Learned

The support of the buying office is critical to the success of the parametric proposal process. A best practice is to involve a buying office's contracting officer or price analyst as early as possible in the work of the implementation team. The team educates the buying office on the estimating techniques being developed/used, and addresses its concerns prior to any proposal submission incorporating those techniques.

The Parametric Estimating Reinvention Laboratory demonstrated that contractors would be more likely to use parametrics as a BOE if Requests for Proposals (RFPs) advocated their use. Appendix C, Frequently Asked Questions, provides an example of a RFP clause which establishes parametric estimating as a valid proposal technique.

Best Practices

The Parametric Estimating Reinvention Laboratory demonstrated that IPTs are a best practice for implementing, evaluating, and negotiating parametrically-based proposals. The Parametric Estimating Reinvention Laboratory also established IPTs at various levels, such as overarching IPTs (OIPTs) and working-level IPTs (WIPTs). Figure J.2 shows IPT team member responsibilities.

OIPTs typically consist of senior management representatives from the contractor, buying activity, DCMA, and DCAA. They are often referred to as Management Councils, and provide a foundation for implementing change, accelerating improvements to the acquisition process, and providing resources to enable the changes. The OIPTs are responsible for chartering multi-functional and multi-organizational items, managing working-level team activities, providing guidance, coordinating issues, resolving disputes, and approving team recommendations. DCMA and DCAA encourage the use of Management Councils.

Effective Management Councils:

- Consist of knowledgeable senior-level people;
- Meet often, with specific agendas;
- Maintain consistent membership;
- Have a well-defined charter.

WIPTs include personnel responsible for process implementation and testing. The WIPT team should include members from the company, the buying activity, DCMA, and DCAA.

Contractor

- Identifies and demonstrates new parametric opportunities.
- Demonstrates how new parametric technique will improve the estimating process.
- Updates estimating system policies and procedures.
- Develops a parametric proposal consistent with defined policies.

Buying Activity, DCMA, and DCAA

- Provide feedback on estimating system disclosure requirements.
- Help to establish acceptance criteria.
- Review calibration and validation results.
- Provide advice on Government evaluation criteria.

Figure J.2 Implementation Team Responsibilities.

Most Parametric Estimating Reinvention Laboratory teams found the IPT process well suited for implementing a parametrically-based estimating system, and developing a parametrically-based proposal. Teams set up subcommittees of subject matter experts to resolve specific issues (e.g., CERs, database validation, regulatory challenges). The Government and contractor more effectively and rapidly resolved problems by addressing these issues at a local, team level. The results demonstrated that everyone gained a better understanding of the parametric tools, and there were fewer surprises during negotiations. The use of an IPT greatly shortens the cycle time from initial RFP to final price agreement.

APPENDIX K

Preparation of Subsystem Level Datasheets

PREPARATION OF

SUBSYSTEM LEVEL DATASHEETS

TABLE OF CONTENTS

LIST OF FIGURES

1.0 GENERAL REQUIREMENTS

Data sheets originally invented by the NRO Cost Group for satellite systems have been adopted by other procurement organizations and are often required input at proposal submittal. This document serves as a guide for providing cost-related data that can characterize a system in the form of data sheets and questionnaires covering hardware, software and costs. They are constantly being updated and modified, so this Appendix is only intended to provide insight into the Data Sheet and its content. **The Data Sheets contained herein are obsolete. The user should ensure he/she has the most current version available by contacting the SPO**.

The following instructions typically apply to cost and cost-related information to be provided for an actual or projected system acquisition contract (referred to hereafter as the "contract"). In some cases involving future systems, the scope of the projected contract may not be completely defined. The term "contract" may refer to the entire projected system or the portions of the system that are defined in other guidance documents. Also, the term "system life cycle" is used to represent situations where acquisition costs may span multiple contracts.

Data is to be provided according to the instructions below. Where two or more satellite, or ground segment configurations are proposed, provide a complete set of data sheets for each configuration.

A. Software Data

Provide descriptive data for all operational software covered by the contract according to the instructions in Section 2. The descriptive data includes software group data sheets and software system development data sheets.

For purposes of this submission, operational software includes support software that is used during the system operational phase, such as simulation software, database management systems, training support software, and fault isolation software. COTS software need not be included in the software data sheets.

B. Ground Segment Hardware Data

Provide descriptive data on ground segment hardware according to the instructions in Section 3. Data is to be provided at the subsystem level or lower, depending on external guidance. All ground segment hardware that is a part of the operational system, both custom and COTS, is to be included in the data sheets unless otherwise specified.

C. Space Segment Hardware Data

Provide descriptive data on space segment hardware according to the instructions in Section 4. Data is to be provided at the subsystem level or lower, depending on external guidance. All space segment hardware that is a part of the operational system, custom, GFE and COTS, is to be included in the data sheets unless otherwise specified.

D. Data Preparation and Submission

Color coding and cell protection are used in the spreadsheets to aid in filling them out. Fields (individual cells or cell groups) for data entry are colored yellow. Blue fields contain values that are calculated from raw input data; these are protected and should not be altered in most instances. Gray fields (protected) indicate that either input data or calculated values are not appropriate. Protected green fields are used for table headings. White areas are protected and are generally for borders and footnotes.

2.0 SOFTWARE DATA

The purpose of this section is to define software technical information. The data requested are structured to be compatible with multiple software cost estimating models. These models include the PRICE S™ model, the Galorath Associates SEER-SEM™ and SEER-SSM™ models, and the COnstructive COst MOdel (COCOMO II) developed by Dr. Barry Boehm. Descriptive material associated with the input parameters used in these models has been freely excerpted from published model documentation, and recognition of their contributions is hereby noted.

Computer Software Configuration Items (CSCIs) and software "data processing groups" are frequently referred to as software end items. The terms "software item" and "item" will be used to denote a CSCI, a software data processing group, or portion of either in subsequent paragraphs. For purposes of this submission, an item should be considered as a functionally related set of computer programs/routines which result from a decomposition of the total deliverable software into smaller sets of software. A given item may contain sections of code written in different source languages (e.g., Ada, C++, FORTRAN, Assembly, Job Control Language).

Software data sheets, an example of which is shown in Figures 2-1a, b, c (three pages), are to be filled out for all items contained in the deliverable software, including existing and/or GFE software. Each sheet covers a set of related software items (e.g., a data processing group), hereafter referred to as a system.

The primary types of data included in the software data sheets are:

- Software item sizes, expressed in terms of Source Lines Of Code (SLOC)
- Data describing the amount of changes and retesting required for existing software items that will be reused
- Complexity Attributes (CAs) characterizing development difficulty for the software items
- Nonrecurring and recurring costs for the total period of the contract.

The size and change/retest data are on the first page, CA information is on the second page and cost data is on the third page. The cost data may include additional lines to cover costs not associated with specific software items, such system engineering, management and integration of related to the items as a whole. The example in Figure 2-1c includes such a line for integration and test of the example subsystem.

Commercial Off-the-Shelf (COTS) software may be included on the third software data sheet (Figure 2-1c) without providing size and change/retest data on the first two pages (Figures 2-1a, b). A row of information is to be filled out for each software item. However, if an item contains code in more than one source language, ***then multiple rows must be filled out for the item, one row per source language.*** This is shown in Figure 2-1a, where software Items 1 and 2 have only one source language (FORTRAN) and Item 3 has two source languages (ASSEMBLY and FORTRAN). A separate row is required only if the amount of code written in the language in question exceeds ten percent of the total lines of code for the item. Languages can be combined on a single line as long as the aggregate does not exceed ten percent.

Instructions for filling out the software data sheets are in Section 2.1. All information for an item must be completely filled out (i.e., data entered in all data columns) except for size and change/retest data on reused software that is not applicable (see Section 2.1).

Some of the CA responses may apply to all items within a system. Likely candidates are included in a system level data sheet, labeled “Software System Development Data” and referred to as a “system data sheet”. An example system data sheet is shown in Figure 2-1d; instructions for filling out system data sheets are given in Section 2.2. The example data is for the case where the data sheets represent an existing system (as opposed to a future system).

If the same CA value applies to all the items in the system that is being described, for the same reason(s), then use the system level data sheet to provide the rationale for the selection of that CA value. Show the value of the CA on the system data sheet and also in the appropriate software data sheet rows, ***followed by an asterisk*** to indicate that its value is supported by a system data sheet (some rows may have CA values that are different from that on the system data sheet).

Figure 2-1a. SOFTWARE DATA SHEET 1 - SIZE DATA EXAMPLE

PREPARER NAME:	J. P. Smith [2]	PHONE #:	Open: (112) 358-1321	Secure: 259-1234 [3]	DATE: 20-Jun-95 [1]
DATA POINT OF CONTACT:	R. S. Jones [4]	POC PHONE #:	Open: (307) 641-4321	Secure: 886-7601 [5]	
WBS#:	03.04.XX [6]	SYSTEM:	Sample Data Processing Group [7]		VERSION: Config 2. Prelim [8]

WBS NO.	ITEM ID	ITEM SIZE DATA [12]																MONTHS SDR TO CSCI TEST	SOURCE OF S/W	CONTR	PERCENT SUBCONTRACT	
		SOURCE LANGUAGE	DELIVERED NEW CODE			PRE-EXISTING CODE [17]																
			UNIQUE KSLOC	COMMON KSLOC	AUTO GEN KSLOC	NOT DESIGNED FOR REUSE [18]						DESIGNED FOR REUSE [25]									1ST TIER	2ND TIER
						TOTAL KSLOC	DELETED KSLOC	CHANGED KSLOC	% RD	% RI	% RT	TOTAL KSLOC	DELETED KSLOC	CHANGED KSLOC	% RD	% RI	% RT					
ITEM FUNCTION: [9]																						
[10]	[11]	[13]	[14]	[15]	[16]	[19]	[20]	[21]	[22]	[23]	[24]	[26]	[27]	[28]	[29]	[30]	[31]	[32]	[33]	[34]	[35]	[36]
ITEM FUNCTION:		Prepares and formats output reports																				
01	FCN1	FORTRAN	6.5	3.0	20.0	26.5	3.0	4.0	5	15	40	0.0	0.0	0.0	0	0	0	20	2	1	30	0
ITEM FUNCTION:		Data processing support functions																				
02	FCN2	FORTRAN	4.5	0.0	0.0	22.5	2.0	3.0	3	13	45	0.0	0.0	0.0	0	0	0	18	1	2	0	0
ITEM FUNCTION:		Primary data processing functions -- assembly language portion																				
03	FCN3	ASSEMBLY	0.5	0.0	0.0	4.0	0.5	0.3	2	8	25	1.0	0.0	0.2	5	20	50	14	1	3	0	0
ITEM FUNCTION:		FORTRAN portion of FCN3																				
03	FCN3	FORTRAN	40.0	0.0	0.0	0.0	0.0	0.0	0	0	0	0.0	0.0	0.0	0	0	70	28	N/A	3	100	0
ITEM FUNCTION:		Subsystem Integration																				
04	I&T																					
ITEM FUNCTION:																						
ITEM FUNCTION:																						
ITEM FUNCTION:																						
ITEM FUNCTION:																						
ITEM FUNCTION:																						

Figure 2-1b SOFTWARE DATA SHEET 2 - COMPLEXITY ATTRIBUTES EXAMPLE

DATE: 20-Jun-95 1

PREPARER NAME: J. P. Smith 2 PHONE #: Open: (112) 358-1321 Secure: 259-1234 3

DATA POINT OF CONTACT: R. S. Jones 4 POC PHONE #: Open: (307) 641-4321 Secure: 886-7601 5

WBS#: 03.04.XX 6 SYSTEM: Sample Data Processing Group 7 VERSION: Config 2. Prelim 8

WBS NO.	ITEM ID	COMPLEXITY ATTRIBUTE RATING (0-10) 37																																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
ITEM FUNCTION:		9																																
10	11																																	
ITEM FUNCTION:																																		
01	FCN1	3	5	0	3	4	3	3	0	5	4	4	0	0	4	0	4	2	2	3	5	5	4	2	3	3	0	0	2	2	2	3	6	5
ITEM FUNCTION:																																		
02	FCN2	3	5	0	2	3	2	4	0	5	4	5	0	0	6	0	7	2	2	0	5	0	5	2	0	3	0	0	2	2	2	4	6	5
ITEM FUNCTION:																																		
03	FCN3	3	5	0	2	3	2	2	0	5	4	8	0	0	6	0	7	2	2	0	5	0	8	2	0	3	0	3	2	5	6	2	4	5
ITEM FUNCTION:																																		
03	FCN3	3	5	0	4	3	4	3	0	5	4	6	0	0	6	0	0	2	2	3	5	5	8	2	4	3	0	3	2	6	6	3	4	5
ITEM FUNCTION:																																		
04	I&T																																	
ITEM FUNCTION:																																		
ITEM FUNCTION:																																		
ITEM FUNCTION:																																		
ITEM FUNCTION:																																		
ITEM FUNCTION:																																		

CA1 -ORGANIZATIONAL INTERFACES
CA2 -SCHEDULE ACCELERATION/STRETCHOUT
CA3 -NEW TARGET HW/OPS SYSTEM (VIRTUAL MACHINE)
CA4 -PERSONNEL EXP. WITH DEV COMP. & SUPPORT SYS.
CA5 -PERSONNEL EXPERIENCE WITH THIS TYPE OF APPLICATION
CA6 -PERSONNEL EXPERIENCE WITH THIS PROGRAMMING LANG.
CA7 -ANALYST CAPABILITIES
CA8 -TARGET SYSTEM VOLATILITY
CA9 -CHANGES TO S/W RQTS (AFTER DESIGN INITIATION)
CA10 -MATURITY OF OPS RQTS DEF. (AT DESIGN INITIATION)
CA11 -COMPLEXITY (S/W DEVELOPMENT DIFFICULTY)
CA12 -IMPACT OF CPU TIME CONSTRAINT ON S/W
CA13 -IMPACT OF CPU/PERIPHERAL MEM. CONSTRAINT ON S/W
CA14 -REQUIRED SOFTWARE RELIABILITY
CA15 -DEV. PERSONNEL & COMPUTER AT REMOTE/OPS SITE
CA16 -SPECIAL DISPLAYS (MMI)
CA17 -SOFTWARE DEVELOPMENT TOOLS
CA18 -TURNAROUND DURING DEVELOPMENT
CA19 -DEV. (HOST) COMPUTER DIFFERENT THAN TARGET
CA20 -USE OF MODERN PROGRAMMING PRACTICES
CA21 -DATA BASE SIZE
CA22 -SOFTWARE INTERFACE COMPLEXITY (INTERNAL)
CA23 -TERMINAL RESPONSE TIME
CA24 -RESOURCE DEDICATION
CA25 -PRACTICES AND METHODS VOLATILITY
CA26 -REUSEABILITY REQUIREMENT
CA27 -SOFTWARE IMPACTED BY REUSE
CA28 -TARGET SYSTEM SECURITY REQUIREMENTS
CA29 -AMOUNT OF REAL TIME CODE
CA30 -SOFTWARE INTERFACE COMPLEXITY (EXTERNAL
CA31 -PROGRAMMER CAPABILITIES
CA32 -PRACTICES AND METHODS EXPERIENCE
CA33 -PRACTICES AND PROCEDURES COMPLEXITY

Figure 2-1c SOFTWARE DATA SHEET 3 - COSTS EXAMPLE

DATE: 20-Jun-95 [1]

PREPARER NAME: J. P. Smith [2] PHONE #: Open: (112) 358-1321 Secure: 259-1234 [3]

DATA POINT OF CONTACT: R. S. Jones [4] POC PHONE #: Open: (307) 641-4321 Secure: 886-7601 [5]

WBS#: 03.04.XX [6] SYSTEM: Sample Data Processing Group [7] VERSION: Config 2. Prelim [8]

WBS NO.	ITEM ID	COST ($M) [38] NON-RECURRING [39]	RECURRING [40]	TOTAL [41]	COMMENTS [42]
ITEM FUNCTION: [9]					
[10]	[11]				
ITEM FUNCTION:					
01	FCN1	3.00	0.80	3.80	2 YEARS MAINTENANCE
ITEM FUNCTION:					
02	FCN2	2.80	0.40	3.20	2 YEARS MAINTENANCE
ITEM FUNCTION:					
03	FCN3	0.40	0.10	0.50	2 YEARS MAINTENANCE
ITEM FUNCTION:					
03	FCN3	4.80	0.60	5.40	2 YEARS MAINTENANCE
ITEM FUNCTION:					
04	I&T	2.70	0.30	3.00	2 YEARS MAINTENANCE
ITEM FUNCTION:					
ITEM FUNCTION:					
ITEM FUNCTION:					
ITEM FUNCTION:					
ITEM FUNCTION:		Subsystem Totals:			
		13.70	2.20	15.90	

2.1 Software Data Sheets

The following descriptions relate to the numbered items on the software data sheet examples (Figures 2-1a, b, and c):

1. **DATE:** The date the sheet was completed.

2. **PREPARER NAME:** The name of the person who gathered the data.

3. **PHONE #:** The phone numbers of the person who gathered the data -- open (nonsecure) and secure.

4. **DATA POINT OF CONTACT:** The name of the person who provided the data.

5. **POC Phone #:** The phone numbers of the person who provided the data -- open and secure.

6. **WBS NO:** The WBS number for the subsystem or processing group.

7. **SYSTEM:** The name of the system or processing group in which this item is located.

8. **VERSION:** The version (preliminary, final, conceptual, etc.) of the item being documented. If multiple versions are being submitted at the same time, then provide a second descriptor identifying which version the sheet represents.

9. **ITEM FUNCTION**: A description of the software item's purpose or function. Include as part of the ITEM FUNCTION description, whether COMMON code (see paragraph 15) is developed as part of this item or included as part of this item.

   Where multiple functions are performed, or where the item is relatively large, include additional information to provide insight into the item's total functionality. Avoid short, ambiguous phrases. Even though the examples in Figure 2-1a are relatively short, **longer, more explicit descriptions are desired.** If additional space is needed, use an extra (blank) sheet for that purpose.

10. **WBS NO.:** The last two WBS digits identifying the WBS element containing the software item.

11. **ITEM ID:** A unique abbreviation used to identify the software item (normally the program or CSCI name).

12. **ITEM SIZE DATA:** The parameters in this block provide a measure of the magnitude or size of the software item. The primary size parameter is Source Lines of Code. Source lines of code do not include blank lines, comments, machine generated or instantiated code (These are to be provided in the AUTO GEN CODE column – see paragraph 16), undelivered test code, undelivered debugging code, or begin statements from begin-end pairs. Source lines of code do include executable source lines such as all control, conditional, mathematical, declaration, input, and output statements, as well as I/O formatting statements, deliverable job control, and debug and test code which is delivered

in the final product, if it is part of the requirements.

The following matrix describes in detail precisely what is and what is not included in a Source Lines of Code count:

For each logical line of code:

Include: All executable lines.
Include: Non-executable declarations and compiler directives.
Exclude: Comments, continuation lines, banners, blank lines, and non-blank spacers.

Also, look at the means by which a line was produced:

Include: Manually programmed lines.
Include: Lines developed by the developer for use as a preprocessor to a Source Code Generator.
Include: Lines converted with automated code translators. However, these lines should be entered as pre-existing code. The user will then define the amount of rework required on the translated code through the use of rework percentages.
Include: Copied, reused, or modified lines of code. Again, these lines should be entered as pre-existing lines of code.
Exclude: Lines generated as output from a Source Code Generator (These are to be provided in the AUTO GEN CODE column – see paragraph 16).
Exclude: Deleted lines of code.

Furthermore, look at the origin of each line:

Include: New lines developed from scratch.
Include: Pre-existing lines taken from a prior version, build, or release.
Include: Invocation statements or lines considered for rework evaluation from COTS or other off the shelf packages. The user should define the level of rework required for those lines which are modified in any way.
Include: Invocation statements only for unmodified vendor supplied or special support libraries.
Include: Modified vendor supplied or special support libraries, commercial libraries, reuse libraries, or other software component libraries. The user should define the level of rework required for those lines which are modified in any way.
Exclude: Lines which are part of an unmodified vendor supplied operating system, utility or other non-developed code.

Lastly, consider the end usage of each line:

Include: Lines which are in or part of the primary product.
Include: Lines which are external to or in support of the primary product, only if

they are part of the final or deliverable program.

Exclude: Lines which are external to or in support of the primary product, but are not deliverable, or any other non-deliverable lines.

Counting Ada Source Lines

Ada source lines are counted as non-comment, non-embedded line terminating semi-colons within the program code, with macros counted only once.

Ada Program Example:

Code Statement	Type
Procedure Example is	Partial
	Blank
-- This is a comment; not line of code	Comment
Type Z is range 4 ..44;	Declaration Source
A:Z: = 22;	Declaration Source
B:Z: = 12;	Declaration Source
Character_literal: CHARACTER: = ';';	Declaration Source
String_literal: STRING: = 'X;'Y";	Declaration Source
Procedure First Is (R: in Z; S: out Z) is separate;	Declaration Source
	Blank
Begin	Partial
if (A=22) then	Partial
B: = 4;	Source
End if;	Source
	Blank
End example;	Declaration Source
Code Count:	
16 Physical lines	
9 Non-Comment, Non-Embedded Semicolons	
4 Comment Statement and Blank	
3 Partial Lines Not Comments or Statements	
9 Ada Source Lines of Code (7 Are Declarations)	

13. **SOURCE LANGUAGE**: The source language in which the software item is written.

14-16. **DELIVERED NEW CODE (KSLOC)**: This block is used to indicate the total number of deliverable source lines of code developed from scratch (designed, coded, and tested) within this item. Size estimates shall EXCLUDE software growth (code growth) provisions or contingencies. The total DELIVERED NEW CODE for the item should be the sum of UNIQUE KSLOC, COMMON KSLOC and AUTO GEN KSLOC.

Software items in which common code is originally developed should include common code sizes in the UNIQUE KSLOC column. Items in which common code is included (reused) should include the sizes of the reused common code in the COMMON KSLOC column and the sizes of the newly developed code in the UNIQUE KSLOC column.

14. **UNIQUE KSLOC**: The total number of UNIQUE new lines of deliverable source code, expressed in thousands. The sum of the KSLOCs for each of the source language rows is equal to the total UNIQUE KSLOC for the software item. Common sections of code (see below) shall be included only once in the UNIQUE KSLOC category, namely in the initial software item which develops them.

15. **COMMON KSLOC**: The total number of COMMON new lines of deliverable source code, expressed in thousands. If a software item contains sections of common code which are identical to those accounted for in the UNIQUE KSLOC of another item, the size of the common code portion of the item is entered here.

16. **AUTO GEN KSLOC**: The total number of new lines of AUTOMATICALLY GENERATED code, expressed in thousands. Enter the size here for code obtained using an automatic code generator. In the comments field (paragraph 42), describe the process and tool(s) used to generate the code.

17. **PRE-EXISTING CODE**: Reused code within the software item falls into two categories: Designed for Reuse (Item 25) and Not Designed for Reuse (Item 18). Each of these has six subcategories, described below.

18. **PRE-EXISTS: LINES NOT DESIGNED FOR REUSE** (Items 19-24)

19. **PRE-EXISTING LINES OF CODE**: Total pre-existing lines (completed before this development) from any source. Size and percent estimates for pre-existing code (TOTAL KSLOC, DELETED KSLOC, CHANGED KSLOC, PERCENT REDESIGNED, REIMPLEMENTED, RETESTED) shall EXCLUDE software growth (code growth) provisions or contingencies.

20. **LINES DELETED IN PRE-EXISTING**: Lines deleted from the total pre-existing software.

21. **LINES CHANGED IN PRE-EXISTING**: Lines changed (rewritten or modified) within the pre-existing software that were not designed for reuse. Do not include new lines of code -- they are accounted for in Item 14, UNIQUE KSLOC. NOTE: This value is used for documentation ONLY and is not used in any computations.

22. **PERCENT REDESIGNED**: The portion of the pre-existing software that requires redesign, reverse engineering, redocumentation, and revalidation to work the new design into the pre-existing software (Item 19). The percentage redesigned can be thought of as the percentage of pre-existing software that required change plus the amount of software that didn’t require change, but did require reverse engineering, redocumentation, or revalidation. This may exceed 100 percent in the case where the pre-existing design was so poorly executed and documented that the effort required to reverse engineer the

software exceeds the effort which would have been required to design the total software subsystem from scratch. The base from which the percentages in Items 22 - 24 are to be calculated is the total lines of pre-existing code which is ultimately reused, i.e. the difference between Item 19, PRE-EXISTING LINES OF CODE and Item 20, LINES DELETED IN PRE-EXISTING.

23. **PERCENT REIMPLEMENTED**: The portion of the pre-existing code that requires reimplementation (coding and unit testing) to make it functional within the software item. INCLUDE amounts of code requiring familiarization, learning, or reverse engineering by the developers responsible for reimplementation, as well as the actual amount of code reimplemented.

24. **PERCENT RETESTED**: The ***effort*** required to test the pre-existing software, ***expressed as a portion of the effort*** that would have been required had the software been developed from scratch. For example, when the percent redesigned and percent reimplemented are relatively low, the percent retested would usually be less than 100 percent because the portions of the software not altered can be retested efficiently, using pre-existing test procedures. On the other hand, the percent retested would normally be higher than the percent reimplemented because (1) the reimplemented portion requires “testing from scratch” (100 percent retested by the above definition) and (2) the rest of the item requires at least testing of interfaces with the reimplemented portion.

25. **PRE-EXISTS: LINES DESIGNED FOR REUSE (Items 26-31)**

26. **PRE-EXISTING LINES OF CODE**: Code within the software item that was developed previously and was originally required to be reusable. This may include reusable component libraries, utility packages, and other software that is integrated into this item. Size and percent estimates for pre-existing code (TOTAL KSLOC, DELETED KSLOC, CHANGED KSLOC, PERCENT REDESIGNED, REIMPLEMENTED, RETESTED) shall EXCLUDE software growth (code growth) provisions or contingencies.

27. **LINES DELETED IN PRE-EXISTING**: Lines deleted from the reusable software during the current development process for this software item.

28. **LINES CHANGED IN PRE-EXISTING**: Lines changed (rewritten or modified) within the pre-existing software. Do not include new lines of code -- they are accounted for in Item 14, UNIQUE KSLOC. NOTE: This value is used for documentation ONLY and is not used in any computations.

29. **PERCENT REDESIGNED**: The amount of redesign (software architecting) that is required to make the pre-existing software (Item 26) functional within the software item. INCLUDE effort required for familiarization, learning, and reverse engineering to make the changes as well as the effort for the actual redesign work. This percentage may be

greater than 100% if the work involved in designing includes severe reverse engineering of poorly documented systems. The base from which the percentages in Items 29-31 are to be calculated is the total lines of pre-existing code which is ultimately reused, i.e., the difference between Item 26, PRE-EXISTING LINES OF CODE and Item 27, LINES DELETED IN PRE-EXISTING.

30. **PERCENT REIMPLEMENTED**: The portion of the pre-existing code requiring reimplementation (coding and unit testing) to make it functional within the software item. INCLUDE amounts of code requiring familiarization, learning or reverse engineering by the developers responsible for reimplementation, as well as the actual amount of code reimplemented.

31. **PERCENT RETESTED**: The effort required to test the pre-existing software, expressed as a portion of the effort that would have been required had the software been developed from scratch (see Item 24).

32. **MONTHS SDR TO CSCI TEST**: The elapsed time (in months) from System Design Review (SDR) to the end of the software item testing (PCA and FCA).

33. **SOURCE OF SOFTWARE**: The name of the software system that serves as the primary source of the reused code (if any). If there is insufficient space available to enter the name, provide a table on a separate sheet indicating numbers and corresponding system names. For example:

No.	Source of Software
1	CSCI ADYN, Project MARS
2	CSCI RACS, Project LEM
3	CSCI LUNA, Project MOON

Enter the number shown in the left column above in field 33.

34. **CONTRACTOR**: The name of the contractor organization responsible for performing the software development. If insufficient space is provided, furnish a table on a separate sheet, indicating numbers and corresponding contractor names. For example:

No.	Contractor Name
1	Galactic Software, Inc.
2	Crown Communications Co.
3	Stellar Satellite Systems

Enter the number in the left column above in field 34.

35,36. **PERCENT SUBCONTRACT**: These data define the amount of effort for the software item which is subcontracted. The first column is the percent of the total effort subcontracted by the prime contractor to first tier subcontractor(s). The second column is the percent of the first tier subcontracted effort performed by second tier subcontractor(s). If the first tier percentage is zero, the second tier percentage must also be zero.

37. **COMPLEXITY ATTRIBUTE (CA) RATINGS**: The 33 attribute ratings in this section of the data sheet relate to the software development environment, personnel experience, operating environment, and overall application difficulty. Guidelines and instructions for completing this portion of the data sheets are provided in Appendix A.

38. This element includes the **NONRECURRING**, **RECURRING** and **TOTAL COSTS** for each software item identified.

    Software costs are classified as follows:

    Space Segment software: all costs associated with the development of the space segment flight software up to and including first launch are defined as NONRECURRING COSTS. Similarly, all costs for the development of test, handling and other ground support and test software should be classified as NONRECURRING COSTS. Costs incurred for maintenance of the software after first launch are defined as RECURRING COSTS.

    Ground Segment software: all costs associated with the development and procurement of ground segment software prior to ground segment Initial Operating Capability (IOC) are defined as NONRECURRING COSTS. If COTS costs are included in the software data sheets, then COTS acquisition costs are classified as NONRECURRING. All costs associated with maintenance of the ground segment software, including annual license fees and upgrades for COTS, are defined as RECURRING COSTS.

    Ground software normally has separate WBS elements for O&M, and they may not necessarily be accounted for at the CSCI level (as is done for the Acquisition Phase). If this is the case, then only the NONRECURRING COSTS column should be filled out for the Acquisition Phase for the applicable WBS elements (CSCIs) and only the RECURRING COSTS column should be filled out for the O&M phase for the comparable software WBS elements.

39. **NONRECURRING COSTS:** Include all costs for development of the software item identified.

40. **RECURRING COSTS:** Include all costs for the maintenance of the software item identified.

41. **TOTAL COSTS:** This is the sum of Non-Recurring and Recurring costs for each software item.

42. **COMMENTS:** Include any additional descriptions of the software items identified, including interfaces, technology, applicability of pre-existing code used, indirect methods of SLOC sizing, systems or item architecture, etc.

FIGURE 2-1d SOFTWARE SYSTEM DEVELOPMENT DATA EXAMPLE

Date: **20-Jun-95** 1

Preparer Name:	**J. P. Smith** 2	**Phone #:**	**Open:**	**(112) 358-1321**	**Secure:**	**259-1234** 3
Data Point of Contact:	**R. S. Jones** 4	**POC Phone #:**	**Open:**	**(307) 641-4321**	**Secure:**	**886-7601** 5
WBS #:	**03.04.02** 6	**System:**		**Sample DP Group** 7		
Version:	**Config 2.Prelim** 8					

System Functional Description: 9
Mission planning software

	CA #	CA Value (0-10) 13
Development Standard: 10 E. DoD-STD-2167A Min **Development Model (Method):** 11 A. Waterfall		
Organizational Interfaces: 12 One SPO, one user, relatively minor communication problems	1	3
Changes to S/W Requirements (after Design Initiation): A number of minor changes made early in the program	9	5
Maturity of Operational Requirements Definition (at Design Initiation): A number of TBDs resolved early	10	4
Development Personnel & Computer at Remote/Operational Sites: Development and support resources immediately at hand	15	0
Software Development Tools: Strong support environment	17	2
Use of Modern Programming Practices: Moderate use of MPPs, evolving	20	5
Practices and Methods Volatility: Gradually evolving practices	25	3
Reusability Requirement: No reusability requirement	26	0
Practices and Procedures Complexity: A substantial change was planned in MDP use	33	5

Host Computer: 14
VAX 11/780

Target Computer: 15
VAX 11/780

Perception of Software Development Task 16
A number of minor operational requirements changes were experienced early in the project, a few beyond CDR. Moderate code growth ocurred, some due to these changes and some due to optimestic estimates.

2.2 Software System Development Data

The following descriptions relate to the numbered items on the example system data sheet in Figure 2-1d:

1. **Date:** The date this form was completed.

2. **Preparer Name:** The name of the person who gathered the data.

3. **Phone #:** The phone numbers of the person who gathered the data -- open (nonsecure) and secure.

4. **Data Point Of Contact:** The name of the person who provided the data.

5. **POC Phone #:** The phone numbers of the person who provided the data -- open and secure.

6. **WBS No:** The WBS number for the system.

7. **System:** The name of the system, subsystem or processing group in which the software items or CSCIs for which data was provided are located.

8. **Version:** The version (preliminary, final, conceptual, etc.) of the system being documented. If multiple versions are being submitted at the same time, then provide a second descriptor identifying which version the sheet represents.

9. **System Functional Description:** Provide a thorough description of the functions which the system or processing group performs. If insufficient space is available, for this or any other data on this page, use extra sheets. If a functional description document has already been developed, then simply attach it.

10. **Development Standard:** From the list of development standards below, select the development standard most like the one employed in this development. Include its description (not just the identification letter).

Development Standard

A. Commercial	I. DoD-STD-1703
B. DoD-STD-2167 (Tailored)	J. MIL-STD-483/490
C. DoD-STD-2167A (Tailored)	K. DoD-STD-1679
D. DoD-STD-2167 (Minimum)	L. DoD-STD-1679 (IV & V)
E. DoD-STD-2167A (Minimum)	M. MIL-STD-SDD
F. DoD-STD-2167 (Full)	N. DoD-STD-498
G. DoD-STD-2167A (Full)	O. ISO 9001
H. IEEE	P. Other______________________

11. **Development Model (Method):** From the list of development methods below, select the development model (method) most like the one employed in this development. Include its description (not just the identification letter). If a combination of more than one are used, select all that apply.

Development Methods

A. Ada Development	H. Traditional Incremental Number of Increments________
B. Ada Dev. With Increments	I. Prototype (Traditional)
C. Ada Full Use	J. Prototype (Rapid)
D. Traditional Waterfall	K. Object Oriented Design/Programming
E. Modified Waterfall	L. Full Object Oriented
F. Spiral	
G. Evolutionary	

12,13. For each of the CAs listed below which apply to the entire system or many of the software items within the system, provide the CA value selected and the rationale for its selection. For those from the list which do not apply uniformly to this system, enter their CAs and rationale individually in the Basis for CSCI Complexity Attributes sheet. (See Figure A-1 and the CA descriptions in Appendix A).

14. **Host Computer:** Describe the host (development) computer(s). Include such data as manufacturer, model, total number of each type used (including redundant ones), performance characteristics, system architecture, reserve requirements, actual reserves, and CSCIs resident simultaneously. If a fact sheet on each type of computer is available, simply attach it.

15. **Target Computer:** Describe the target computer(s). Include such data as manufacturer, model, total number of each type used (including redundant ones), performance characteristics, system architecture, reserve requirements, actual reserves, and CSCIs resident simultaneously. If a fact sheet on each type of computer is available, simply attach it.

16. **Perception of Software Development Task:** For future systems, describe characteristics of the system software development project that provide insight relative to its inherent difficulty, the current degree of requirements "firmness" and depth of definition, the ability to predict final software item sizes, and assumptions implicit in the software and system data sheet responses.

    For existing systems, describe how the perception of the software development task changed over time. For example, what was learned about the requirements and development environment over the life of the project which would give rise to a changed perception of the nature of the task or the technical attributes of the software? This question applies to software size, as well as other attributes.

3.0 GROUND SEGMENT HARDWARE DATA SHEET

Name: ________________________________ Date: ________________

Organization: _________________________________ Phone:_________________

System Name: ______________________________________

This segment is included for completeness. Only fill these out if your design requires ground segment hardware changes. The information collected herein will be used to prepare cost estimates on the ground segment. For each Major Ground Element in Table 1, provide descriptive data as instructed in the Ground Segment Hardware Data Description shown in Figure 3-1. Complete a separate data sheet for each major ground element identified in the table below.

Table 1. Major Ground Elements

Major Ground Element	Description
Mission Planning	Mission planning, mission control, and collection plan execution functions for the satellites and ground stations
Command and Control	All hardware and facilities required to generate and upload commands to the satellite
Satellite Communications	All hardware and facilities required to receive and process state of health telemetry/tracking and mission data
Mission Data Processing	All payload processing and computer hardware and facilities required for the processing of raw mission data
Analysis and Reporting	All hardware and facilities required for analysis of processed payload data and generation of intelligence reports
Dissemination	All communications and computer hardware and facilities for dissemination to exploitation and analysis units
Other External Communications	All communications and computer hardware and facilities for establishing communication links to external nodes and assuring the quality of those links. These functions include terrestrial communications network management, mission support, and tactical communications
Common Services	Ground site system engineering, integration and test, management, and administrative support. This element also includes ancillary support, such as system simulations and facility support

Provide a track between major ground elements and the ground segment work breakdown structure (WBS). Include all aspects of each major ground element (communications and computer hardware, facilities, etc.) with the exception of ground segment software. Ground software should be described using the Software Data Sheets (Figures 2-1a, b, c) and Software System Development Data Sheet (Figure 2-1d) discussed in Section 2, Software Data.

Provide descriptive information on the hardware end items associated with each hardware element and head counts for labor-related elements. Where practical, break out head counts by function, such as engineering, operations, maintenance, or management. If a particular item is not applicable, please so indicate with N/A.

Figure 3-1. Ground Segment Hardware Data Description

Provide the following descriptive information for each Major Ground Element:

Briefly describe the space element type/function that the ground element supports (e.g., LEO/MEO/HEO/GEO satellite command and control, signal/imagery data collection, electro-optical/radar imaging, etc.)

Number of ground stations to be developed, modified, produced, or purchased?

What is the uplink frequency (ies) (GHz)?

What is the downlink frequency (ies) (GHz)?

Is the ground system mobile, transportable, or a fixed site?

If the system is mobile or transportable, will it be ruggedized and will it be capable of transmitting/receiving while moving?

Location(s) of the deployed system(s).

Mission of the system (e.g., TT&C, communications, data processing, all)?

Type of satellite tasking system to be utilized.

Location(s) of the satellite command and control facility.

Describe the type and identify the number of new and existing hardware end items (antennas, transmitters, amplifiers, receivers, recording devices, encrypt/decrypt units, etc.) required to perform each major function. As an example, when describing a ground antenna, provide antenna diameter, type of feed and modes, frequency range, and efficiency at center frequency. Briefly describe the development effort including number of prototypes and, at the subsystem level (or lower level, if available) the percentage of redesign/design required for modifications to existing equipment. Identify assumptions for commercial off-the-shelf (COTS) item usage.

Identify percentage of hardware commonality throughout the ground system (e.g., are all workstations the same type).

List all new facilities (buildings and vans, power generation and distribution systems, backup power systems, system timing and time code generators, heating and air conditioning, and security) required. Identify major modifications required to existing facilities.

Identify the total number of military, civilian and contractor personnel at each ground facility.

4.0 SPACE SEGMENT HARDWARE DATA SHEET

Name: ________________________________ Date: _________________

Organization: _________________________________ Phone:_________________

System Name: _____________________________________

The information collected herein will be used to prepare cost estimates on the space segment. Provide descriptive data for all deliverable hardware in the operational system(s), including GFE and purchased or subcontracted hardware. Provide a block diagram of the bus and payload. Items not in the operational system such as test equipment and ground support hardware do not have to be described. There are two sets of data sheets that are required. The first set is described under Hardware Data Sheets and the second under Supplementary Hardware Data.

Hardware Data Sheets (Figure 4-1) are to be used to list the separate hardware items that make up functional subsystems. These components may be mechanical or structural systems, subsystems, assemblies and/or electronic boxes. Optimally, each row in the form should be used for an individual assembly or box. For large assemblies which break down physically into smaller assemblies, a separate row should be used for each smaller assembly.

Complex mechanical devices with moving parts (such as deployment mechanisms and gimbals) should be separated from other types of mechanical/structural components such as equipment compartments, booms and pallets. Wiring harnesses and cables interconnecting electronic boxes should be listed separately from electronic boxes (and from other mechanical items). Assembly A in Figure 1 illustrates the segregation of different types of hardware; each type (subassembly) is on a different row. When program maturity permits the following guidance applies. For space hardware, each individual electronic box should be on a separate line. For ground hardware, one line per drawer is desired for custom rack-mounted equipment.

In some cases, a single line can be used to represent a collection, or set, of relatively low cost items. This approach is preferred over the use of multiple lines (one for each item). The items included in such a line need not be identical, but they should be similar in complexity. The following are specific cases for satellites where this approach is desirable:

- Brackets, fasteners and other miscellaneous structural items
- Cable harnesses - all cable harnesses should be collected in one or two lines unless there is a specific reason to use more lines. The use of two lines, one for bus harnesses and one for payload harnesses, is preferred over collection of both types in one line.
- Propulsion valves, sensors and propellant lines (plumbing) should be combined into one line. Relatively large manifold assemblies should be listed on a separate line but inclusion with valves, sensors and lines is acceptable.
- Microwave plumbing

As a general rule, lines containing collections of smaller items should not represent more than 20 percent of the total subsystem cost. However, somewhat larger percentages are acceptable for cable harnesses and propulsion plumbing.

The following describes the data to be entered in the Hardware Data Sheets (see identification numbers in circles on the sample data sheet in Figure 4-1):

1. **WBS NO.:** The WBS number for the subsystem, integrated assembly or equipment group.
2. **SUBSYSTEM:** The name of the subsystem, integrated assembly or equipment group.
3. **VERSION:** (Conceptual, preliminary, final, etc.) The version of the configuration being documented. If multiple versions are being submitted at the same time, then provide a second descriptor identifying which version the data sheet represents.
4. **WBS NO**: The last two WBS digits identifying the assembly or box at the next lower WBS level (box/assembly level).
5. **DESCRIPTION:** The name or description of the assembly or box.
6. **UNIT WEIGHT:** The total weight of the assembly or box, *including margin for weight growth.* Where the line is being used to represent N identical items (boxes or assemblies), the unit weight is the weight of *one box or assembly.* Where the line represents a collection of different items, the unit weight is the *sum of the weights* of a11 of the items represented.
7. **% MARGIN:** The percent growth margin which is included in the total unit weight, where the percent margin is calculated with the *unmargined weights as the denominator.*
8. **NO. PER SYS.:** Number of units per system. For lines representing collections of different items, the number per system would be 1.
9. **WT. PER SYS:** Total weight per system for these units. This is the product of the unit weight (item 7) and the number of units per system (Item 10).
10. **HERITAGE:** Identify the heritage ranking of each item with respect to the NASA Maturity Index below:
11. **COMMENTS:** Provide justification for chosen heritage as well as any additional necessary comments
12. **POWER PER SYS:** Power per unit x the number of active units

NASA Maturity Index

Ranking	Heritage Description	Heritage	Benchmarks
A, B	Totally off-the-shelf. All hardware may be readily purchased with minimum modifications and qualification.	80-100%	0.0-0.1
C	Basic design exists, minor modifications	60-70%	0.3-0.4
D	Basic design exists, requires significant modifications (40-50% new or modified boxes/assemblies)	40-50%	0.5-0.6
E	Similar design exists, major modifications (mostly new or modified boxes/assemblies).	20-30%	0.7-0.8
F	Nominal new design. Some advanced development has been accomplished.	0-10%	0.9-1.0
G	Requires new design, development, and qualification. Little or no advanced development work has been accomplished.		1.1

Figure 4-1

DATE: ______ (1)

PREPARER:_______________

WBS #: 01.02.XXX

PHONE:___________________

VER. CONFIG 2 PRELIM (3)

SUBSYSTEM: SAMPLE HARDWARE SUBSYSTEM (2)

SHEET NO.:________________

WBS NO.	UNIT DESCRIPTION	UNIT WEIGHT WT-LB	UNIT WEIGHT % MR	NO. PER SYS	WT PER SYS	PWR PER SYS	HERITAGE	COMMENTS
4	5	6	7	8	9	12	10	11
02	ASSY A				215	130		
	SUBASSY A-1	134	25	1	134	60	C	PARTIAL NEW DES; NO ENG MODEL PQ
	SUBASSY A-2	18	15	2	36	30	D	PARTIAL NEW DES
	SUBASSY A-3	45	10	1	45	40	D	PARTIAL NEW DES
03	ASSY B	36	20	1	36	32	F	NEW DEVELOPMENT; MOSTLY UNIQUE
04	ASSY C	16	35	6	96	50	B	REPACKAGE EXISTS
	SUBSYSTEM TOTAL				347	212		

Supplementary Hardware Data Format –

1. Satellite

- Design Life (Months) ____________
- Operating Life After Deployment (Months) ____________
- Satellite Dry Weight (Kilograms) ____________

2. Structure

- Structure Material Composition (% by weight)
  Provide a percentage mix of materials used in the structure

  Aluminum ____________
  Magnesium ____________
  Stainless Steel ____________
  Simple Alloys ____________
  Fiberglass ____________
  Titanium ____________
  Beryllium ____________
  Boron/Graphite Composite ____________
  Hybrid ____________
  Other (Specify______________________) ____________

3. Attitude Determination and Control Subsystem

- List Sensor Types (e.g., sun, star, horizon. etc.)________________________________
- Number of Axes Stabilized ____________
- Design Life of Reaction Control System (Months) ____________
- Type of Stabilization (Spinner, 3-Axis, etc.) ____________

4. Electrical Power Supply

- Beginning of Life Power (Watts) ____________
- Number and Type of Solar Cells (Si, GaAs, Ge, etc.) ____________
- Battery Capacity (Amp Hours per Battery) ____________
- Number of Batteries ____________
- Type of Batteries (NiCd, NiH, NaS, etc.) ____________

5. Apogee Kick Motor (if applicable)

- Type of AKM (Model, Manufacturer) ____________________________________
- Total Impulse (Kg/sec) ____________
- AKM Dry Weight (Kilograms) ____________

6. Telemetry, Tracking, & Command / Communication Payload Subsystems

Supply the following data for each applicable TT&C or communication payload subsystem. Appropriate definitions are found in #11 below.

- Number of Channels in the Subsystem ____________

7. RF Transmitters, Receivers

For each transmitter/receiver provide: quantity in the subsystem, output power (watts), frequency range of operation, number of channels, commonality with boxes in other subsystems on this satellite, whether solid state or TWTA, and for TWTA transmitters, the TWTA weight.

8. Signal Processor

Provide the following data for the signal processor component:

- Quantity in System ____________
- Number of Links ____________
  (see #11 below - differs for TT&C vs. communication application)
- Box Function ____________
  (Processing, Data Handling, General Elex, Telemetry, Freq Synth. Receiver.)
- Design Complexity ____________
  (Low, Medium or High)
- Box Volume (cu in) ____________ Minimum Likely ____________ Most Likely ____________ Max Likely
- Installed Power (W) ____________ Minimum Likely ____________ Most Likely ____________ Max Likely
- Number of Technologies in Box:
  - GaAs (% for each occurrence of this technology in the box)

	Tech 1	Tech 2	Tech 3	Tech 4	Tech 5	Tech 6
Frequency (GHz)						
1st Yr of Manufact.						

  - Si RF (% for each occurrence of this technology in the box)

	Tech 1	Tech 2	Tech 3	Tech 4
Frequency (GHz)				
1st Yr of Manufact.				

  - SAW ____________
    (None, Crystal, Digital)
  - VHISIC ____________
    (Yes or No)
  - Digital ____________
    (Yes or No)
- Boards in Modules ____________

(Yes or No)

9. Antennas

- Number of antenna systems ____________

For <u>each</u> antenna provide as applicable: solid dish - (Y/N), directional or omni, reflector diameter, effective area in square feet, weight without support structure, and commonality with antennas in other subsystems on this satellite.

10. RF Distribution

- Is the Distribution System active (Y/N)? __________
- If there is an active portion, give the weight of the active portion. __________
- What is the waveguide weight? __________

11. PARAMETER EXPLANATIONS

Effective Area. Effective Area is based on the antenna gain and frequency. This area is more properly called the electromagnetic cross sectional area of the antenna and is calculated according to the formula:

Effective Area = $\lambda^2 * 10^{gain\_db/10}$

where λ = wavelength in feet

gain_db = the antenna gain in decibels

Design Life. Design life is the length of the orbital operational period for which the satellite is designed, expressed in months.

Number of Antenna Systems. A single antenna system may contain a number of antenna elements. For example, if an antenna system is made up of five identical helical elements, it would be counted as one antenna system (i.e. it would be designed and developed as a unit). It also is common for more than one (i.e. two or four) feed horn to be grouped to form one antenna system.

Number of Channels. Number of channels equals the total number of transmit/receive channels within the entire communication subsystem.

Number of Digital Electronic Boxes. Number of digital electronic boxes is the total number of individual boxes that have digital output and are integrated into the subsystem.

Number of Links. Number of links is determined differently for the TT&C and communication subsystems. For TT&C, it is determined by counting the independent transmit channels supported by the TT&C subsystem. For example, if a satellite has a separate transmit link during the boost phase and is switched to another one for orbital operations, report this as two links. The links associated with non-communication payloads should not be counted. For the communication subsystem, the number of links is determined by the number of processor boxes. Most communication subsystems have one integrated processor box to support several links. In these cases one link should be reported. However, some communication subsystems support several links through several processors. In these cases report the number of processor boxes.

Number of Solar Cells. Number of solar cells is the total of all individual solar cells attached to the solar array panels and/or satellite body.

APPENDIX X-1

SOFTWARE COMPLEXITY ATTRIBUTE INSTRUCTIONS

COMPLEXITY ATTRIBUTE (CA) RATINGS

The 33 attribute ratings in this section of the data sheet relate to the software development environment, personnel experience, operating environment, and overall application difficulty. Figure A-1 provides an outline of the attributes with brief guidelines for appropriate responses. The paragraphs below discuss each attribute in greater detail.

The complexity attribute ratings are to be completed for each programming language within a software item. In those cases where existing code is modified, the ratings should address only those portions (e.g., modules or sub-items) which are modified (i.e., the portions designated in paragraphs 21 and 28) rather than all of the code. Naturally the new code and the modified code together form this base. The attribute ratings should be established based on the average conditions (development environment) prevailing during the development phase.

Note that throughout the complexity attribute descriptions, the term experience refers only to the total accumulation of periods of time (expressed in years) during which the attribute under consideration (e.g., computer system, application, language experience) was actually being addressed by the individuals involved. For example, if a programmer worked steadily with the C language for two years in his first programming assignment, and then was reassigned to a programming task where a different language was used for four years, the programmer’s experience with the C language would be 2 years (not 6 years).

FIGURE X-1. COMPLEXITY ATTRIBUTES GUIDELINES

CA1 - ORGANIZATIONAL INTERFACES

0 = One small group, no external organizational interfaces
1 = Single organization, project oriented. A few small task groups
2 = Efficient interfaces between working groups, modularized approach
3 = Only minor problems in coordination and communications
4 = Moderate coordination and communication problems
5 = Development at two locations or several organizational interfaces
6 =
7 = Development at two locations and several organizational interfaces
8 =
9 = Development at 5 or more separated sites or complex organizational interfaces
10 = Development at 5 or more separated sites and complex organizational interfaces

Factors:

- Number of internal task groups/divisions, number of subdivisions, number of customer/user organizations
- Efficiency of internal project management structure
- Effect of geographical separation as it impacts project communications

Note: Technical interfaces are addressed by CA22 and CA30

*Usually the same for most or all CSCIs (i.e., project-level attribute)

CA2 - SCHEDULE ACCELERATION/STRETCHOUT

0= Time to development completion; 75% of optimal development schedule
1 = 80% of optimal completion date
2 = 85%
3 = 90%
4 = 95%
5 = Optimal development schedule (100%)
6 = 115%
7 = 130%
8 = 145%
9 = 160%
10 = Greater than 175% of optimal completion date

CA3 - NEW TARGET HARDWARE/ OPERATING SYSTEM (VIRTUAL MACHINE)

0 = Widely used and mature target computer system
1 =
2 = Mature target computer system, not used widely
3 =
4 = Modified target H/W or O/S
5 =
6 = New computer OR new operating system
7 =
8 = One new, one partially new (revised)
9 =
10= New computer AND new operating system

CA4 - PERSONNEL EXPERIENCE WITH DEVELOPMENT COMPUTER AND SUPPORT SYSTEMS

0 = More than 4 years average experience
1 =
2 = 3 years average experience
3 =
4 = 2 years average experience
5 =
6 = 1 years average experience
7 =
8 = 4 months average experience
9 =
10 = Less than 4 months average experience

Note: Response based on development team average

CA5 - PERSONNEL EXPERIENCE WITH THIS TYPE OF APPLICATION

0 = More than 10 years average experience or reimplementation by the same team
1 =
2 = 7 years average experience
3 =
4 =
5 = 3 years average experience
6 =
7 =
8 = 1 year average experience
9 =
10 = Less than 4 months experience

Note: Response based on development team average

FIGURE X-1. COMPLEXITY ATTRIBUTES GUIDELINES (Continued)

CA6 - PERSONNEL EXPERIENCE WITH THIS PROGRAMMING LANGUAGE
0 = More than 4 years average experience 1 = 2 = 3 years average experience 3 = 4 = 2 years average experience 5 = 6 = 1 year average experience 7 = 8 = 4 months average experience 9 = 10 = Less than 4 months average experience **Note**: Response based on development team average

CA7 - ANALYST CAPABILITIES
0 = Near Perfect Functioning Team (90th percentile) 1 = 2 = Extraordinary (80th percentile) 3 = 4 = Functional and Effective (60th percentile) 5 = 6 = 50th percentile 7 = 8 = Functional with Low Effectiveness (30th percentile) 9 = 10 = Non-Functioning Team (15th percentile) Note: Response based on development team

CA8 – TARGET SYSTEM VOLATILITY
0 = No H/W development 1 = 2 = 3 = Small amount of H/W development, localized impact on S/W 4 = 5 = Some overlaps of development. Most H/W available for testing S/W and vice versa 6 = 7 = Much overlap, little H/W available for testing 8 = 9 = 10 = Simultaneous development, separate organizations, etc.

CA9 - CHANGES TO S/W REQUIREMENTS (AFTER DESIGN INITIATION)
0 = No changes 1 = 2 = 3 = Very few changes expected 4 = 5 = Minor changes to requirements caused by design reviews or changing mission requirements 6 = 7 = Some significant changes expected (none late in development phase) 8 = 9 = 10 = Expect major changes occurring at different times in development phase **Factors** • Frequency of changes • How late in the development cycle changes occur • Maturity of developing SPO, experience of SPO personnel • Number, maturity and influence of user/funding organizations • Stability of economic and political environment (funding stability)

CA10 - MATURITY OF OPERATIONAL REQUIREMENTS DEFINITION (AT DESIGN INITIATION)
0 = No effect 1 = All requirements under configuration control 2 = 3 = Minor number of TBDs, with little effect on design 4 = 5 = Many TBDs, but few with major impacts 6 = 7 = Major TBDs, with moderately severe impacts 8 = 9 = 10 = Operational requirements shifting, feasibility studies incomplete

FIGURE X-1. COMPLEXITY ATTRIBUTES GUIDELINES (Continued)

CA11 – COMPLEXITY (SOFTWARE DEVELOPMENT DIFFICULTY)
0 = Real time not an issue. Extremely simple S/W with primarily straightforward code, simple I/O, and internal storage arrays.
1 =
2 = Background processing. Computational efficiency has some impact on development effort. S/W of low logical complexity using straightforward I/O and primarily internal data storage.
3 =
4 = New standalone system developed on firm operating system. Minimal I/F problems.
5 = Typical C&C
6 = Minor real time processing, significant logical complexity, some changes to operating system.
7 =
8 = Challenging response time requirements, new system with significant I/F and interaction requirements (e.g., OS and R/T with significant logical code)
9 =
10 = Extremely large volumes of data processing in short time, signal processing system with extremely complex I/Fs (e.g., parallel processing, microcode applications).

CA12 - IMPACT OF CPU TIME CONSTRAINT ON SOFTWARE
0 = 50% CPU power still available during maximum utilization
1 = 45% CPU power still available
2 = 40% CPU power still available
3 = 35% CPU power still available
4 = 30% CPU power still available
5 = 25% CPU power still available
6 = 20% CPU power still available
7 = 15% CPU power still available
8 = 10% CPU power still available
9 = 7% CPU power still available
10 = 5% CPU power or less still available

CA13 - IMPACT OF CPU OR PERIPHERAL MEMORY CONSTRAINT ON SOFTWARE
0 = Greater than 50% reserve of memory available
1 = 45% reserve available
2 = 40% reserve available
3 = 35% reserve available
4 = 30% reserve available
5 = 25% reserve available
6 = 20% reserve available
7 = 15% reserve available
8 = 10% reserve available
9 = 7% reserve available
10 = Additional memory must be provided

CA14 - REQUIRED SOFTWARE RELIABILITY
0 = No requirement
1 = Minor inconvenience – easily recoverable
2 =
3 =
4 =
5 = Moderate recoverable loss
6 =
7 = High financial loss
8 = Mission loss
9 =
10 = Failure results in loss of human life

CA15 - DEVELOPMENT PERSONNEL & COMPUTER AT REMOTE/OPERATIONAL SITE
0 = 100% access to development and support resources
1 =
2 =
3 = 70% access
4 =
5 =
6 =
7 = 40% access
8 =
9 =
10 = Less than 25% access

CA16 - SPECIAL DISPLAYS (MMI)
0 = No displays
1 =
2 =
3 = A few simple displays
4 =
5 = User friendly error recovery and menus, character based, window formats, color
6 =
7 = Interactive: Touch screens, light pens, mouse, etc. Controlled by the computer program (Graphics based – 1990's)
8 =
9 = High human-in-the-loop dependencies. Many interactive displays, monitors or status outputs
10 = Complex requirements. CAD/CAM Solid Modeling. Many interactive displays or status outputs (e.g., real time alarms)

FIGURE X-1. COMPLEXITY ATTRIBUTES GUIDELINES (Continued)

CA17 - SOFTWARE DEVELOPMENT TOOLS

0 = Automated full Ada APSE
1 =
2 = Fully integrated application development environment
3 =
4 = Modern (Ada Min. APSE & design, requirements, or test tools)
5 =
6 = Interactive (Programmer Work Bench)
7 =
8 = Basic batch tools (370 OS type (compiler, editor))
9 =
10 = Primitive tools (Bit switches, dumps)

CA18 - TURNAROUND DURING DEVELOPMENT

0 = Interactive, never any waiting
l =
2 = Interactive. Constant availability. Dedicated development machine.
3 =
4 =
5 = Turnaround 4 hours
6 =
7 =
8 = Off-line machine. Tightly scheduled usage. Turnaround greater than 12 hours
9 =
10 = No dedicated machine. Development shared with operations

CA19 – DEVELOPMENT (HOST) COMPUTER DIFFERENT THAN TARGET

0 = Development on target computer
1 =
2 =
3 = Minor changes
4 =
5 = Small amount of reconfiguration required to run on target
6 =
7 = Significant language and systems differences between development (host) and target computer
8 =
9 =
10 = Major alterations of S/W required for testing/checkout

CA20 - USE OF MODERN PROGRAMMING PRACTICES

0 = Maximum benefit of MPPs realized
1 =
2 =
3 = General use of MPPs by personnel experienced in their use
4 =
5 = Some use of MPPs by personnel experienced in their use
6 =
7 =
8 = Beginning experimental use of MPPs
9 =
10 = No use of MPPs

Factors:

- Experience of development team with MPPs
- Extent to which management commits to MPPs

CA21- DATA BASE SIZE

0 = No data base
1 = Data base size/SLOC = 5
2 = Data base size/SLOC = 10. Data easily managed. Requirements/structure known.
3 =
4 = Data base size/SLOC = 30
5 = Data base size/SLOC = 55. Nominal data base size. Not access bound nor other critical constraint
6 =
7 = Data base size/SLOC = 450
8 = Data base size/SLOC = 1000. High access/ performance requirements
9 =
10 =

CA22 - SOFTWARE INTERFACE COMPLEXITY (INTERNAL)

0 = Single module (no external interfaces)
1 =
2 = Loosely coupled items, minor timing constraints
3 =
4 =
5=Closely coupled interfaces. Strict timing protocols, many interrupts
6 =
7 = Numerous or complex interfaces
8 =
9 =

	10 = Tight coupling and strictest timing protocols and constraints

FIGURE X-1. COMPLEXITY ATTRIBUTES GUIDELINES (Continued)

CA23 - TERMINAL RESPONSE TIME

0 = $\leq$.25 second
1 =
2 = .5 second
3 =
4 =
5 = 1.0 second
6 =
7 =
8 = 2.0 seconds
9 =
10 = $\geq$ 3.0 seconds

CA24 – RESOURCE DEDICATION

0 = 100% fully dedicated computing resources
1 =
2 =
3 = 70% access to computing resources
4 =
5 =
6 =
7 = 30% access to computing resources
8 =
9 = 10% access to computing resources
10 =

CA25 - PRACTICE AND METHODS VOLATILITY

0 = No major changes, minor change each year
1 =
2 = Major change each 12 months, minor each month
3 =
4 =
5 = Major change each 6 months, minor each 2 weeks
6 =
7 =
8 = Major changes each 2 months, minor each week
9 =
10 = Major change each 2 weeks, minor 2 times a week

CA26 - REUSABILITY REQUIREMENT

0 = No reusability requirement
1 =
2 =
3 = Software designed for reuse within a single application area (single contractor; multiple/single customers)
4 =
5 =
6 =
7 = Software designed for reuse with a single product line (multiple contractors; multiple/single customers)
8 =
9 =
10 = Mission software developed with full reusability required. All components of the software must be reusable.

CA27 - SOFTWARE IMPACTED BY REUSE

0 = 0% reusable
1 =
2 =
3 = 30% reusable
4 =
5 =
6 =
7 = 70% reusable
8 =
9 =
10 = 100% reusable

CA28 - TARGET SYSTEM SECURITY REQUIREMENTS

0 = Class D: minimal protection - no security
1 =
2 = Class C 1: (see instructions for class definitions)
3 =
4 = Class C2
5 =
6 = Class B1
7 =
8 = Class B2
9 = Class B3
10 = Class A1

FIGURE X-1. COMPLEXITY ATTRIBUTES GUIDELINES (Concluded)

CA29 - AMOUNT OF REAL TIME CODE	**CA32 – PRACTICES AND METHODS EXPERIENCE**
0 = 0% source lines with real time considerations 1 = 2 = 3 = 30% of source lines with real time considerations 4 = 5 = 6 = 7 = 70% of source lines with real time considerations 8 = 9 = 10 = 100% of source lines with real time considerations	0 = More than 4 years average 1 = 2 = 3 years average experience 3 = 4 = 2 years average experience 5 = 6 = 1 year average experience 7 = 8 = 4 months average experience 9 = 10 = Less than 4 months experience **Note**: Response based on development team average
CA30 – SOFTWARE INTERFACE COMPLEXITY (EXTERNAL)	**CA33 – PRACTICES AND PROCEDURES COMPLEXITY**
0 = Single module (no external interfaces) 1 = 2 = Loosely coupled items, minor timing constraints 3 = 4 = 5 = Closely coupled interfaces. Strict timing protocols, many interrupts 6 = 7 = Numerous or complex interfaces 8 = 9 = 10 = Tight coupling and strictest timing protocols and constraints	0 = No change in Modern Development Practices (MDP) use from the established development 1 = 2 = 3 = Moderate change – Organization improving development technologies equivalent to a 1 level transition on the MDP practices use rating 4 = 5 = 6 = 7 = Major change – Organization improving development technologies equivalent to a 2 level transition on the MDP practices use rating 8 = 9 = 10 = Extreme change – Organization improving development technologies equivalent to a 3 level transition on the MDP practices use rating
CA31 – PROGRAMMER CAPABILITIES	
0 = Near Perfect Functioning Team (90th percentile) 1 = 2 = Extraordinary (80th percentile) 3 = 4 = Functional and Effective (60th percentile) 5 = 6 = 50th percentile 7 = 8 = Functional and Low Effectiveness (30th percentile) 9 = 10 = Non-Functioning Team (15th percentile) **Note**: Response based on development team average	

CA1: Organizational Interfaces

This attribute addresses the complexities in coordination engendered by the project organization. Numerous organizational interfaces can slow the progress of development. Here "organizational interfaces" not only refers to those between development organizations (e.g., joint ventures, subcontractors, or divisions within the same company), but also to those between the developers and the procuring organizations (e.g. customers, users, IV&V, etc.). The assessment of this attribute should be tempered by such considerations as the modularity (i.e., number of interfaces) of the software assignments and the efficiency of the organizational interfaces. List the organizations involved in the space provided.

CA2: Schedule Acceleration/Stretchout

Indicates how the actual development schedule relates to an optimal schedule. Only the development phase is considered here. Optimal is to be interpreted in terms of manpower utilization (i.e., highest efficiency). The midpoint (CA value of 5) represents an optimal schedule. A value below 5 implies a project accelerated to meet an early deadline; a value above 5 implies more time is available than needed.

CA3: New Target Hardware/Operating System (Virtual Machine)

This attribute addresses the newness of the target hardware for the development, i.e., how long it has been in general use. "Bugs" can arise in new hardware to impede progress. Documentation, such as user manuals, may be in need of refinement. The intermediate values provide latitude in which to adjust the assessment based on the degree of newness or extraneous considerations.

CA4: Personnel Experience With Development Computer and Support Systems

This attribute measures the amount of experience that the development team assigned to the software item has with the host computer and support systems (operating system, job control languages, database management system, etc.). The rating is based on the **average duration of experience** for the team members. The duration should be increased (i.e., CA4 value decreased) to adjust for experience on very similar systems. Rate the experience level ***at the start of the project***.

For example, a new project begins with 7 programmers, a lead programmer with 10 years of experience with the development system hardware and operating system being used, 3 programmers with 2 years experience each, and the other 3 with no applicable experience with this system. The average experience would be about 2 years. Therefore, from Figure A-1, a value of 4 would be specified for CA4.

CA5: Personnel Experience With This Type of Application

Indicate the **average** duration of experience the development team has with this type of application. The duration should be increased (i.e., CA5 value decreased) to account for experience with very similar applications.

This complexity attribute addresses the analyst team's relevant experience in designing similar applications. Rate the team's experience when design begins. For example, if a new program was started with 3 analysts, 2 of which were fresh out of college, and a third with 10 years experience, the average experience would be about 3 years, or a nominal 5 on the CA5 rating scale in Figure A-1.

CA6: Personnel Experience With This Programming Language

Indicate the **average** duration of experience the development team has with the particular computer language for the software item (one data sheet line per language). The duration should be increased (i.e., CA6 value decreased) to account for experience with similar languages. Rate experience at the beginning of the software development project.

This complexity attribute represents the programming team's average experience with the programming language being used. For example, a new project begins with 7 programmers, a lead programmer with 10 years of experience with the language being used, 3 programmers with 2 years experience each, and the other 3 with no applicable experience with this language. The average experience would be about 2 years, so a value of 4 would be specified for CA6.

CA7: Analyst Capabilities

Rates the analyst team (not individuals) assigned to the software item development. Use the percentile rating scale shown in Figure A-1.

Analysts include personnel developing software requirements and specifications and preparing high-level software design (architecture) normally prepared before Preliminary Design Review. A nominal team is characterized as quite respectable, performing at an average level. Performance may be impacted by inherent learning abilities, efficiency, motivation, quality of design, communication abilities, etc. Conflicts within a team and an uncooperative environment can also reduce this complexity attribute rating. The capability level should be evaluated based on the group as a team rather than as an aggregate of individuals.

Capabilities should not be confused with experience. The Analyst Capabilities attribute rates the inherent potential of the individual team members as well as the team as a whole, independent of experience. More experienced personnel are not necessarily more capable, and less capable personnel are also not necessarily less experienced.

CA8: Target System Volatility

This attribute is directed toward new or changing target computers, support systems, and hardware with which this software item interacts. When the target hardware is still undergoing development, the hardware/software interfaces are likely to change during the course of software item development. Hardware errors are also likely to occur. It may be difficult to distinguish these errors from software errors.

Indicate the extent of further hardware development occurring after the start of the software development. If the hardware systems to be used have been fully developed no effect (CA value of 0) should be indicated. Also consider the impact of the target system operating system and support software when establishing the value for CA8, thinking in terms of how much additional development effort is caused by target system immaturity (relative to the impact of target system hardware and/or support systems immaturity or unavailability).

CA9: Changes to Software Requirements (After Design Initiation)

The factor addressed here is the volatility of the software requirements after software design begins. Requirement changes, for example, can result from modification of the mission objectives, adoption of another operating system or language, or redirection in the software design approach. Minor changes may include work such as software subsystem specification clarification or a user interface menu. Moderate changes are items such as tighter performance requirements. Major changes are items such as rework of major system specifications related to mission changes.

A CA9 value of 10 represents a very high degree of volatility, a situation where significant amounts of redesign and reprogramming are needed to accommodate major change. A value of 5 represents a nominal case, i.e., a typical amount of changes. In the space provided, list the requirements changes which had major impact on the effort expended on this part of the software.

CA10: Maturity of Operational Requirements Definition (at Design Initiation)

The focus here is on the status, at the time of design initiation, of requirements definition for the system and/or operations the software was intended to support. The assessment should account for the following factors: (1) the level of detail to which the operational requirements are defined, (2) the number of gaps (TBDs) in the definition, and (3) the articulation of the requirements in terms of clarity and specificity. The evaluation should be limited to those operational requirements which have a bearing on the design of the software under consideration. In the space provided, describe the maturity of operational requirements definition at design initiation, noting percent completion of relevant documents.

CA11: Complexity (Software Development Difficulty)

This attribute is an indicator of the rate at which personnel can be added to the software development project. It rates the software item's inherent difficulty. For complex projects, engineering problems can often only be solved by small groups of people working in an iterative or sequential manner, and therefore the overall rate that staff can be added to the project is constrained.

Complexity relates to the rate at which personnel can be added to a development program, thus it affects both cost and schedule. A more complex task is more difficult to work on. Consequently, the number of people working on that task must be reduced. However, when the staff size is reduced, there may be less effort lost to the inefficiencies of communication and team integration, so an actual cost benefit can be achieved for more complex tasks, although the schedule length will at the same time be increased.

For example, the task of computerizing the phone book is a large but simple task. Many people could work on it at the same time, simply dividing it into equal size chunks and letting each team member work on a much smaller piece of the problem. The schedule would thus be accelerated. On the other hand, a very complex application might impose a constraint on the number of people who can work on it, thus lengthening the schedule although perhaps decreasing costs.

Additionally, schedule constraints can override this particular setting. In other words, stretching the schedule will cause staff to be added to a project more slowly than what is dictated by this parameter, thus forcing the project to work with smaller, more efficient teams and take longer to complete while saving effort overall.

Select the complexity category in Figure A-1 that best matches the type of system and development requirements for this software item.

CA12: Impact of CPU Time Constraint on S/W

This attribute is measured in terms of the degree of execution time constraint imposed upon the software item. The rating is expressed in terms of the percentage of available CPU execution time that is used by the item **and any other items consuming the execution time resource at the same time**.

Even though this attribute is measured by percent of available memory reserve, it is intended to reflect the effort required by developers for specific coding to enhance timing performance. Thus, if no special coding to improve speed is required, a value of 0 would be used, even if the available processing power margin eventually approaches zero. This might be the case where a software application controls the volume of processing input so that the processor capacity is used efficiently, using additional processors to handle the total processing load.

CA13: Impact of CPU or Peripheral Memory Constraints on S/W

This attribute is measured in terms of the degree of CPU and/or peripheral memory constraints imposed upon the software item relative to its data storage requirements. CPU memory refers to random access memory (RAM). Other devices such as drums, disks, tapes, or bubble storage are also included to the extent their capacity limits affect software development. The rating is based on the percentage of available data storage used by the item **and any other items consuming the storage resource at the same time**.

Even though this attribute is measured by percent of available memory reserve, it is intended to reflect the effort required by developers to reduce memory usage. If no action is required by the development team to conserve memory, then a value of 0 would be used, even when the available memory is 99% utilized. This might be the case where an application's memory usage is "sized" to intentionally use virtually all of the memory available so that capability or processing speed is maximized. It might also be the case where the item storage requirements grew "gracefully" over the course of the development, ending near the hardware limits.

CA14: Required Software Reliability

Indicate the degree of reliability required for the software as imposed by the mission objectives and operating environment. A software product possesses reliability to the extent that it can be expected to perform its intended functions satisfactorily. The ratings are based on the consequences of failure. A CA value of 1 represents a minor inconvenience to the developers to fix the fault. A CA value of 5 represents a moderate loss to the users, but a situation from which one can recover without extreme penalty. A CA value of 10 represents the possible loss of human life. The degree of required reliability generally does not vary from module to module within a subsystem.

CA15: Development Personnel and Computer at Remote/Operational Site

This attribute is intended to account for problems associated with software development away from the home office and support services. If the software is developed at the home office a value of zero should be assigned. Otherwise, a comparison should be made with the case where nominal development resources are available at the home base. Three factors should enter into the evaluation: (1) contention for the development computer (i.e., the extent to which it is dedicated to this project), (2) access to support resources (e.g., documentation, clerical assistance, telephones, supplies), and (3) travel requirements. Extraneous considerations may enter here to mitigate the access problems.

For instance, development at the operational site may allow the project team to interface with cognizant personnel or draw on resources (e.g., documentation) related to the operational system. A low CA value indicates that very few problems are imposed by the remote development, whereas a high CA value indicates severe problems. Use the space provided to explain the circumstances which formed the basis for the numerical response.

CA16: Special Displays (MMI)

This attribute rates the amount of effort required to implement user interface display interaction involved with this computer program. If the software item has no user interface, such as when the interfaces are with other computer programs, CA16 would be assigned a value of 0.

As an example, a Windows-based application with a simple menu-driven interface would be assigned a value of 5. Even though the application would accept mouse input, in this case the mouse interaction is handled by the operating system and is "free" to the application. However, if the application also includes advanced usage of mouse features directly, a value of 7 or 8 might be used.

CA17: Software Development Tools

Indicate the degree to which the software development practices have been automated and are used on this development. The degree of integration of the support tool environment should also be taken into consideration.

The following lists of tools and the general criteria shown in Figure A-1 can be used to establish CA17:

Very Low Level of Automation, CA17=10 (1950s era)

- Assembler
- Basic Linker
- Basic Batch Debug Aids
- High Level Language Compiler
- Macro Assemble

Low Level of Automation, CA17=7 (1960s era)

- Overlay Linker
- Batch Source Editor
- Basic Library Aids
- Basic Data Base Aids
- Advanced Batch Debug Aids

Nominal Level of Automation, CA17=5 (1970s era)

- Multi-User Operating System
- Interactive Source Code Editor
- Data Base Management System
- Basic Data Base Design Aids
- Compound Statement Compiler
- Extended Overlay Linker
- Interactive Text Editor
- Simple Program Design Language (PDL)
- Interactive Debug Aids
- Source Language Debugger
- Fault Reporting System
- Basic Program Support Library
- Source Code Control system
- Virtual Memory Operating system
- Extended Program Design Language

High Level of Automation, CA17=2 (1980s era)

- CASE Tools
- Basic Graphical Design Aids
- Advanced Text Editor (Word Processor)
- Implementation Standards Enforcer
- Static Source Code Analyzer
- Program Flow and Test Case Analyzer
- Full Program Support Library w/CM Aids
- Full Integrated Documentation System
- Automated Requirements Specification & Analysis
- General Purpose System Simulators
- Extended Design Tools and Graphics Support
- Automated Verification System
- Special Purpose Design Support Tools such as display formatters, cross-compilers, communications processing tools, data entry control tools, conversion aids, etc.
- RDBM’s

Very High Level of Automation, CA17=0 (1990s era)

- Integrated Application Development Environment
- Integrated Project Support Environment
- Visual Programming Tools
- Automated Code Structuring
- Automated Metrics Tools
- GUI Testing Tools
- 4GLS
- Code Generators
- Screen Generators

CA18: Turnaround During Development

This attribute measures the system throughput time experienced by the project team. The following actions are included:

- Logon if not dedicated developer terminal (Includes travel and waiting time)
- Invoking an editor
- Submit a software unit for compilation
- Receive hard (or soft) copy into hands

One hundred percent access to the development resources is assumed (i.e., no lockout of resources due to other software developments) during the entire logon through hard copy response cycle.

This parameter is designed to measure the development system throughput time. The ratings in Figure A-1 are defined in terms of the average response time (i.e., turnaround) in hours. Response time here is defined as the time from job submittal by the developer to receipt of results.

CA19: Development (Host) Computer Different Than Target

This attribute addresses the situation where the target computer (i.e., the final operational system) differs from the computer system on which the software was developed. It reflects the effort to convert the software from the development system (computers, operating systems, etc.) to the target system.

If the two computers are identical, assign a zero value to CA19. Otherwise, assign a value to reflect the degree of difference between the two systems. The degree of difference should be evaluated based on the amount of redesign and recoding required in rehosting onto the target system.

CA20: Use of Modern Programming Practices

Rate the usage of modern software development practices and methods ***at the time the software design begins*** using the scale in Figure A-1. These include structured or object oriented methods, development practices for code implementation, documentation, verification and validation, data base maintenance, and product baseline control. The following is a list of modern programming/development practices:

- Object Oriented Design
- Object Oriented Programming
- Incremental Development
- Rapid Prototyping
- Empirically Guided Software Development
- Integrated Life-cycle Application Development
- Integrated Product Teams
- Concurrent Engineering
- Object and Code Reuse Libraries
- Structured Requirements Analysis
- Structured Design
- Structured Programming
- Program Design Language (PDL)
- Incremental Development
- Design/Code Walkthroughs/Inspections
- Team Development Strategies
- Project Estimating and Control

Only successful incorporation of practices as standard procedures within the organization as well as by the team responsible for development of this software item counts as full use. Just having books, a few experts, or attending academic courses does not count as experience. Use the space provided to indicate the specific practices used on this software development.

CA21: Data Base Size

Indicate the size of the database associated with the item. If no database is used, assign a CA value of zero. The ratings are here defined in terms of the ratio of the database size (in terms of bytes or characters) to the item size (in terms of source lines of code). The database size refers to the amount of data to be assembled and stored in peripheral storage (e.g., tapes, disks, drums, bubble memories) by the time of software acceptance.

CA22: Software Interface Complexity (Internal)

This attribute addresses the difficulty in integrating and testing the components to the CSCI level. The scope of interfaces here is restricted to interaction with lower level software items (e.g., CSCs in the case where a CSCI is being reported). The evaluation should be made relative to similar systems (of this application type). A nominal value (5) represents a typical level of interface requirements. Values less than five describe simpler interfaces, perhaps characterized by loosely coupled items. Values greater than five describe interfaces of greater complexity than average for this type of software.

CA23: Terminal Response Time

This attribute rates the average transaction response time from the time a developer presses a key until that key is acknowledged and its action is completed. This measures the efficiency of interactive development operations.

CA24: Resources Dedication

This CA refers to availability of computer resources to the development staff during the software development activity. It rates the availability of the host and target machines to the development organization. Physical interference due to site operations or contending project organizations (on a data processing facility operated by a separate organization) can result in reduced access to the system. Additionally, the sharing of scarce hardware resources can lower resource dedication if the developers are actually locked out.

CA25: Practices and Methods Volatility

This attribute represents the frequency of changes to the processes, methods, and tools that establish the managerial and technical environment in which the software item is developed (e.g., design reviews, quality assurance activities, CASE tools). The rating scale in Figure A-1 refers to the track history of the organization responsible for developing the software item, as this history may be used to infer or project future volatility.

This rating is dependent on the scope or magnitude of the changes as well as the frequency with which they occur. A minor change would be any change that would have some impact on the development team, but would not require significant adjustments to the way in which they work. For example, filling out an additional form or consulting an additional management source for a design decision approval would be a minor change. A major change would require a significant adjustment in the way in which the development team works, and would have a noticeable impact on the development effort.

CA26: Reusability Level Required

This attribute rates the requirements for producing software that is reusable within other programs/applications. Reusable code is defined as code which is fully usable with no modifications (i.e., reusable as-is).

The level of reusability required is partially determined by how widely the final software will need to be reused. For example, code designed to be reused in future builds or incremental releases of a single application would have a relatively low CA26 value (say 1-3), while code designed to be reused throughout an office automation product line (word processors, spreadsheets, databases, etc.) would be have a much higher CA26 value, say 7-8.

This input is used in conjunction with CA27, Software Impacted By Reuse, which measures how much of the total code will be reused.

CA27: Software Impacted by Reuse

This attribute measures the amount (fraction) of the software under development that is required to be reusable.

CA28: Target System Security Requirements

This CA refers to the security requirements included within the delivered application software. It does NOT refer to the physical security required for the development environment or locale. This attribute rates development impacts of security requirements for the delivered target system (all classifications are identified in the Orangebook and described below). Rate special work performed during this item development only.

> Class A1: Security formally verified by mathematical proof. (Only a few known systems)
>
> Class B3: System excludes code not essential to security enforcement. Audit capability is strengthened. System almost completely resistant to penetration.
>
> Class B2: System segregated into protection critical and non-protection critical elements. Overall system resistant to penetration. (Critical financial processing)
>
> Class B1: In addition to C2, data labeling and mandatory access control are present. Flaws identified by testing are removed. (Classified or financial transaction processing)
>
> Class C2: Users individually accountable via login operations, auditing of security relevant events and resource isolation. (Typical VAX operating system such as VMS)
>
> Class C1: Access limited. Based on system controls accountable to individual user or groups of users. Simple project specific password protection.
>
> Class D: Minimal protection--no security.

If security requirements are met by the operating system, by other software, or by a physically secure environment (behind locked doors), then the value of CA28 should be 0.

Higher security levels can be EXTREMELY expensive to implement with software; it is usually more cost effective to meet these requirements with physical security than with software.

CA29: Amount of Real Time Code

This attribute rates the amount of software directly involved in real-time functions. Real time functions are driven by a clock external to the software, such as gathering data from hardware devices or time sensitive control of such devices where waiting can alter or lose data. Real time functions must be performed during the actual time that an external process occurs in order that the computation results can be used to control, monitor, or respond in a timely manner to the external process. Real-time code manages data exchange across the interfaces, but not the processing of data (which is in non-real-time). For example, telemetry is gathered in real-time, but can be processed in non-real time. (NOTE: Although Real Time Code is not directly related to the CPU Time Constraint described in CA12, some code may require timing constraints because of real time considerations).

CA30: Software Interface Complexity (External)

This attribute addresses the difficulty in integrating and testing the CSCIs to the system level. The scope of interfaces here is restricted to interaction with other software items (say CSCIs, if that is the level to which the software development is being reported). The evaluation should be made relative to similar systems (of this application type). A nominal value (5) represents a typical level of interface requirements. Values less than five describe simpler interfaces, perhaps characterized by loosely coupled items. Values greater than five describe interfaces of greater complexity than average for this type of software.

CA31: Programmer Capabilities

Rate the average intrinsic capability of the programming team responsible for programming the software item. Use the percentile rating scale shown in Figure A-1.

Programmers perform “code to” detailed design, (program design languages, flowcharts, etc.) write code and prepare and run initial unit test cases. Rate the team, not just the individuals. Consider inherent ability, motivation, programming efficiency and thoroughness, program quality, and the ability to communicate within the development team.

As with analysts (CA7), capabilities should not be confused with experience. This parameter rates the inherent potential of the individual team members as well as the team as a whole, independent of experience. More experienced personnel are not necessarily more capable, and less capable personnel are also not necessarily less experienced.

CA32: Practices and Methods Experience

This attribute rates the **average** software development team experience with the development methods (such as object oriented design), standards (such as DoD-Std-2167A), and procedures (e.g., design reviews, quality assurance activities) applicable to this development. Rate experience ***as of the beginning of the development project.***

CA33: Practices and Procedures Complexity

This attribute evaluates the impact of improving by comparing the established software processes currently in use with those planned for the development of this software item. Software processes are those processes, methods, and tools that establish the managerial and technical environment in which software products are developed (e.g., design reviews, quality assurance activities, CASE tools).

For example, if a development organization had customarily used little or no modern development practices, but on a new project they planned to implement several modern development practices, including Object Oriented Design, Object Oriented Programming, and Concurrent Engineering, they would be rated as relatively low on the Modern Development Practices scale (CA20), which is where they would be at the start of the development project. However, the Practices and Procedures attribute CA33 would be a 7, corresponding to a two-level change on the Modern Development Practices Scale (CA20). This is equivalent to a two-level jump in the Software Engineering Institute process rating scale.

Glossary

List of Acronyms

3GL	Third Generation Language
AAF	Adaptation Adjustment Factor
AAM	Adaptation Adjustment Multiplier
ABC	Activity Based Costing
ACAP	Analyst Capability
ACES	Advanced Cost Estimating System
ACO	Administrative Contracting Officer
ACT	Annual Change Traffic
AFCAA	Air Force Cost Analysis Agency
AFP	Adjusted Function Points
ANOVA	Analysis of Variance
APEX	Applications Experience
ASIC	Application-Specific Integrated Circuit
B&P	Bid and Proposal
BFP	Basic Function Points
BOE	Basis of Estimate
BOM	Bill of Material
C2	Command and Control
C3	Command, Control, and Communications
C3I	Command, Control, Communications, and Intelligence
CAD/CAM	Computer Aided Design & Manufacturing
CAIG	Cost Analysis Improvement Group
CAIV	Cost as An Independent Variable

CALFAC	Calibration Factor
CAM	Contract Audit Manual
CAS	Cost Accounting Standards
CASE	Computer-Aided Software Engineering
CCDR	Contractor Cost Data Report
CER	Cost Estimating Relationship
CES	Cost Element Structure
CIP	Continuous Improvement Process
CMM	Capability Maturity Model
CMMI	CMM Integration (Models)
CMMI-SE/SW	CMMI for Systems and Software Engineering
CMU	Carnegie Mellon University
COCOMO	Constructive Cost Model
COMPEAT$	Cost Offering Method for Affordable Propulsion Engineering Acquisition and Test
COTS	Commercial Off-the-Shelf
CPI	Consumer Price Index
CPLX	Product Complexity
CSCI	Computer Software Configuration Item
CV	Coefficient of Variation
DATA	Database Size
DAU	Defense Acquisition University
DCAA	Defense Contract Audit Agency
DCMA	Defense Contract Management Agency
DDT&E	Design, Development, Test & Evaluation
DEM/VAL	Demonstration/Validation
DFARS	Defense Federal Acquisition Regulation Supplement
DoD	Department of Defense
DSI	Delivered Source Instructions
DTC	Design-to-Cost
E&MD	Engineering and Manufacturing Development

EAC	Estimate of Completion
EAF	Effort Adjustment Factor
EBS	Estimating Breakdown Structure
EDP	Electronic Data Processing
EM	Effort Multiples
ERP	Enterprise Resource Planning
ESC	Executive Steering Committee
ESS	Error Sum of Squares
ETC	Estimate to Completion
ETR	Effective Technology Rating
EVMS	Earned Value Management System
FAR	Federal Acquisition Regulation
FFRDC	Federally Funded Research and Development Center
FLEX	Development Flexibility
FPRA	Forward Pricing Rate Agreement
G&A	General and Administrative
GAO	General Accounting Office
GUI	Graphical User Interface
HCA	Head of the Contracting Activity
HOL	Higher Order Language
HWCI	Hardware Configuration Item
IC	Integrated Circuits
IDIQ	Indefinite Delivery Indefinite Quantity
IFPUG	International Function Points User’s Group
IPAO	Independent Program Assessment Office
IPT	Integrated Product Team
IR&D	Independent Research and Development
IS	Information Systems
ISO	International Organization for Standardization
ISPA	International Society of Parametric Analysts
IT	Information Technology

JEROP	Joint Estimating Relationship Oversight Panel
KDSI	Thousands of Delivered Source Instructions
KSLOC	Thousands of Source Lines of Code
LANG	Programming Language
LCC	Life Cycle Cost
LCCE	Life Cycle Cost Estimate
LEXP	Programming Language Experience
LOC	Lines of Code
LOE	Level-of-Effort
LRIP	Low-Rate Initial Production
LRU	Line Replaceable Unit
LSBF	Least Squares Best Fit
LTEX	Language and Tool Experience
MAF	Maintenance Adjustment Factor
MCPLXE	Manufacturing Complexity of Electronics
MIS	Management Information System
MODP	Modern Programming Practices
MOU	Memorandum of Understanding
MRP	Materials Requirements Planning
MSFC	Marshall Space Flight Center
MTBF	Mean Time Before Failure
MTTR	Mean Time to Repair
NAFCOM	NASA/Air Force Cost Model
NAP	New Application Points
NASA	National Aeronautical and Space Administration
NASCOM	NASA Cost Model
NEWC	New Code
NEWD	New Design
NOP	New Objects Point
NUC	Normalized Use Care
OIPT	Overarching Integrated Product Team

OSD	Office of the Secretary of Defense
PCAP	Programmer Capability
PCEI	Parametric Cost Estimating Initiative
PCO	Procurement Contracting Officer
PCON	Personnel Continuity
PDIF	Platform Difficulty
PERS	Personnel Capability
PERT	Program Evaluation and Review Technique
PEXP	Platform Experience
PMAT	Process Maturity
PNM	Price Negotiation Memorandum
POP	Predictive Object Point
PPI	Producer Price Index
PREX	Personnel Experience
PRICE H	PRICE Hardware Model
PRICE S	PRICE Software Model
PVOL	Platform Volatility
QSM	Quantitative Software Management
R	Coefficient of Correlation
R2	Coefficient of Determination
RCPX	Product Reliability and Complexity
RDT&E	Research Development Test and Evaluation
REDSTAR	Resource Data Storage and Retrieval
RELY	Reliability Requirements
RESL	Degree of Risk Resolution
REVIC	Revised Enhanced Version of Intermediate COCOMO
RFP	Request for Proposal
ROI	Return on Investment
ROM	Rough Order of Magnitude
ROTC	Reduced Total Ownership Costs
RUSE	Reusability Requirements

SC	Software Component
SCE	Software Capability Evaluations
SCEA	Society of Cost Estimating and Analysis
SCI	Software Configuration Item
SEE	Standard Error of the Estimate
SEER-DFM	Software Evaluation and Estimation of Resources Design-for-Manufacturability Tool
SEER-H	Software Evaluation and Estimation of Resources Hardware Model
SEER-HLC	Software Evaluation and Estimation of Resources Hardware Life Cycle Model
SEER-IC	Software Evaluation and Estimation of Resources Integrated-Circuit Model
SEER-SEM	Software Evaluation and Estimation of Resources Software Estimating Model
SEER-SSM	Software Evaluation and Estimation of Resources Software-Sizing Model
SEI	Software Engineering Institute
SER	Size Estimating Relationship
SF	Scale Factors
SICAL	Size Calibration Factor
SITE	Multiple Site Development
SLOC	Source Lines of Code
SPA	Software Process Assessment
SPR	Software Productivity Research
SRU	Sub Replaceable Unit
SSCAG	Space Systems Cost Analysis Group
SU	Software Unit
TCI	Total Cost Input
TDEV	Development Time
TIME	Execution Time Constraints
TINA	Truth in Negotiations Act
TOC	Total Ownership Cost

TURN	Computer Turnaround Time
UFP	Unadjusted Function Points
UML	Unified Modeling Language
USA	United States Army
USAF	United States Air Force
USC	University of Southern California
USD (AT&L)	Under Secretary of Defense for Acquisition and Technology and Logistics
USN	United States Navy
UTIL	Utilization
VEXP	Virtual Machine Experience
VHOL	Very High Order Language
VIRT	Virtual Machine Volatility
VOL	Volume
WBS	Work Breakdown Structure
WG	Working Group
WIPT	Working-Level Integrated Product Team
XH	Extra High

Definition of Terms

Algorithmic Methods — Use of mathematical formulas to make estimates. The formulas are derived from research and historical data and use inputs that relate to physical characteristics, performance characteristics, functional characteristics, productivity factors, and other known cost driver attributes. Parametric models are considered algorithmic models.

Analogy Methods — A comparison of a new program to a completed one (for which the cost and schedule are known) by deriving factors to relate the costs based on similarities between the programs.

Analysis — A systematic approach to problem solving. Complex problems are simplified by separating them into more understandable elements, which may be solved individually or in combination.

Anomalies	A variance in cost or noncost (technical) data caused by an unusual event(s) that is not expected to recur in the future.
As Spent Dollars	The cost of a project, in real year dollars, without normalization for inflation.
Audit	The systematic examination of records and documents to determine the adequacy and effectiveness of budgeting, accounting, financial, and related policies and procedures; to determine the compliance with applicable statutes, regulations, policies and prescribed procedures; and to determine the reliability, accuracy, and completeness of financial and administrative records and reports.
Audit Trail	Information allowing the data used to develop an estimate to be tracked back to the original source for verification.
Bottoms-Up Methods	A method of estimating characterized by a thorough, detailed analysis of all tasks, components, and assemblies. The results are "rolled-up" to produce an estimate of the entire project. Bottoms-up estimates can be developed using a combination of many estimating methods. Other names for bottoms-up methods are detailed estimating and grass roots estimating.
Calibration	A process of adjusting cost models and relationships to a specific contractor's historical cost experience and business culture. Calibration is accomplished by calculating adjustment factor(s) to compensate for differences between an organization's historical costs and the costs predicted by the cost models and relationships.
Coefficient of Determination (R2)	A measure of the strength of the association between the independent and dependent variables (x and y). The range of R2 is between zero and one. An R2 of zero indicates that there is no relationship between x and y. An R2 of one indicates there is a perfect relationship between x and y.
Computer-Aided Software Engineering (CASE) Tools	Tools that aid software engineers during one or more phases of the software life cycle (e.g., analysis, design, programming, and testing). CASE tools can be used individually to support specific life cycle activities such as identifying software requirements, or developing documentation. CASE tools can also be integrated to support a majority of the software life cycle activities.
Configuration Item	Hardware or software, or an aggregate of both, which is designated for separate configuration management. Configuration items may vary widely in complexity, size,

	and type. Examples of configuration items are aircraft, electronics, and payroll information systems.
Configuration Management	A discipline applying technical and administrative controls to the identification and documentation of physical and functional characteristics of configuration items; the identification and documentation of changes to characteristics of those configuration items; and the recording and reporting of change processing and implementation of the changes to the system.
Constant Dollars	Computed values that remove the effects of inflation and show all dollars at the value they would have been in a selected base year.
Correlation	A statistical technique used to determine the degree to which variables are related or associated. Correlation does not prove or disapprove a cause and effect relationship.
Cost Analysis	The accumulation and analysis of actual costs, statistical data, and other information on current and completed contracts or groups of contracts (programs). Cost analysis also includes comparisons and analyses of these data, and cost extrapolations of data for future projections of cost. An example of cost analysis is the review and evaluation of an organization's cost or pricing data and of the judgmental factors applied in projecting from the data to the estimated costs. Cost analysis is performed to establish an opinion on the degree to which the organization's proposed costs represent what the performance of the contract should cost, assuming reasonable economy and efficiency.
Cost-Benefit Analysis	A technique for assessing the range of costs and benefits associated with a given option. A criterion for comparing programs and alternatives, when benefits can be valued in terms of dollars or costs. Cost-benefit analysis is useful in the search for an optimal program mix, which produces the greatest number of benefits over costs.
Cost Drivers	The characteristics of a system or item having a major effect on the system's or item's cost. Characteristics are defined as a distinguishing feature or attribute. Characteristics can be functional, physical, or operational. Functional characteristics include performance parameters such as range, reliability, and speed. Physical characteristics are tangible in nature and include items such as weight, volume, and shape. Operational characteristics relate to functions that will be performed by

	the equipment such as type of modulation, frequency, and channeling.
Cost Effectiveness	The magnitude of the benefits to be derived from a system with cost as the primary or one of the primary measures.
Cost Effectiveness Analysis	A method for examining alternative means of accomplishing a desired military objective or mission. The purpose is to select the alternative that will provide the greatest cost effectiveness.
Cost Estimating	The act of approximating the cost of something based on information available at the time. Cost and price estimating is the art of predetermining the most realistic cost and price of an item or activity, including normal profit.
Cost Estimating Relationships (CERs)	Mathematical expressions of varying degrees of complexity expressing cost as a function of one or more cost driving variables. CERs can utilize cost-to-cost variables or cost-to-noncost variables. An example of a cost-to-cost CER is manufacturing hours based on quality assurance hours. An example of a cost-to-noncost CER is engineering hours based on the number of engineering drawings.
Cost Factor	A CER in which cost is directly proportional to a single independent variable such as a percentage of labor or material.
Cost Model	A compilation of cost estimating logic that aggregates cost estimating details into a total cost estimate. An ordered arrangement of data, assumptions, and equations that permits translation of physical resources or characteristics into costs. Cost models generally consist of a set of equations, logic, programs, and input formats. Cost models also allow for the input of program information, including both system description data and estimating relationships. Cost models generally produce a variety of output formats.
Cost or Pricing Data	All facts that, as of the date of price agreement or, if applicable, an earlier date agreed upon between the parties that is as close as practicable to the date of agreement on price, prudent buyers and sellers would reasonably expect to affect price negotiations significantly. Cost or pricing data requires certification in accordance with Federal Acquisition Regulation (FAR) 15.406-2. Cost or pricing data are factual, not judgmental; and are verifiable. Cost or pricing data are all facts that can be reasonably

	expected to contribute to the soundness of estimates of future costs and to the validity of determinations of costs already incurred.
Cost Realism	A demonstration that the costs in an offeror's proposal are realistic for the work to be performed; reflect a clear understanding of the requirements; and are consistent with the various elements of the offeror's technical proposal.
Delivered Source Instructions (DSIs)	The number of source lines of code developed for a software project. The number of DSIs is a primary input to many software estimating tools. The term "delivered" excludes non-delivered support software such as test drivers. The term "source instructions" includes all program instructions created by project personnel that progress into code, excluding comments and unmodified utility software. DSIs include job control language, format statements, and data declarations.
Delphi Technique	A technique for applying the informed judgment of a group of experts, using a carefully planned program of sequential individual issues, without direct confrontation; and with maximum use of feedback of digested information in the investigation and solution of problems.
Design-to-Cost (DTC)	An acquisition management technique used to achieve system designs that meet stated cost requirements. Cost, as a key design parameter, is addressed on a continuing basis and is an inherent part of the development and production process. Rigorous cost goals are established during development and the control of the system's costs (e.g., acquisition, operations, and support) to these goals is achieved by practical tradeoffs among operational capabilities, performance, cost, and schedule.
Embedded	A computer system physically incorporated into a larger system whose function is not data processing. While embedded resources may be stand-alone, they must be integrated with a larger system. Embedded computer resources can be used for other purposes, provided their primary function is to support a weapons system.
Enterprise Resource Planning (ERP)	An Industry term for the broad set of activities supported by multi-module application software that help a manufacturer or other company manage the important parts of its business, including product planning, parts purchasing, maintaining inventories, interacting with suppliers, providing customer service, and tracking orders. ERP can also include application modules for the finance

	and human resource aspects of a business. Typically, an ERP system uses, or is integrated with, a relational database system. The deployment of an ERP system can involve considerable business process analysis, employee retraining, and new work procedures.
Equivalent Units	The number, or fraction, of completed units at a given time.
Expert Judgment Methods	Uses the experience and understanding of human experts to develop the project estimates. An advantage of this method is the experience from past projects that the expert brings to the proposed project.
Fixed Costs	Costs that do not vary with the volume of business. Examples include depreciation and property taxes.
Forward Pricing Rate Agreement (FPRA)	A written agreement negotiated between a company and the Government to make certain rates available during a specified period for use in pricing contracts or modifications. Such rates represent reasonable projections of specific costs that are not easily estimated for, identified with, or generated by a specific contract, contract end item, or task. These projections may include rates for such things as labor, indirect costs, material obsolescence and usage, and material handling.
Historical Data	Data reflecting actual costs or past experience of a product or process. Historical data includes all recorded information for a contract. Data can be either technical information or non-technical (e.g., financial).
Improvement Curve	The continuous reduction of the unit values incurred (i.e., labor hours) of end items in the manufacturing process. The improvement curve may result from more efficient use of resources, employee learning, new equipment and facilities, and/or improved flow of materials. When reflecting only labor hours the curve is generally called a learning curve (defined below).
Information Other Than Cost or Pricing Data	Any type of information that is not required to be certified in accordance with Federal Acquisition Regulation (FAR) 15.406-2 and is necessary to determine price reasonableness or cost realism. Information other than cost or pricing data may include pricing, sales, or cost information.
Integrated Product Teams (IPTs)	A composition of representatives from all appropriate functional disciplines working together to build successful processes and make sound and timely decisions. IPTs

	function in a spirit of teamwork with participants empowered and authorized, to the maximum extent possible, to make commitments for the organization or functional area they represent.
Knowledge Base	The repository of knowledge in a computer system or organization. The collection of data, rules, and processes that are used to control a system, especially one using artificial intelligence or expert system methods.
Learning (or Improvement) Curve	A tool of calculation used primarily to project resource requirements, in terms of direct manufacturing labor hours, or the quantity of material required for a production run. A mathematical way to explain and measure the rate of change of cost as a function of quantity. The learning curve is based on the assumption that as a quantity of units produced is doubled, the value declines by a constant percentage. The constant rate of decline is the slope of the learning curve, which is linear when plotted on a loglog scale. Several different learning curve theories exist and are utilized within the Industry.
Life Cycle	The total life span of a system, commencing with concept formulation and extending through operation and eventual retirement of the system.
Management Council	A team of senior representatives involved with business activities at a particular contractor facility. Management Councils used on DoD procurements generally include representatives from the contractor, major customers, Defense Contract Management Agency (DCMA), and Defense Contract Audit Agency (DCAA). They provide a forum to discuss, coordinate, and resolve issues of common concern affecting efficiency and effectiveness of contractor operations and facilitate coordination of business and manufacturing process re-engineering initiatives.
Management Information Systems (MIS)	An orderly and disciplined accounting and reporting methodology, usually mechanized, which provides for the accurate recording of data, and the timely extrapolation and transmission of management information used in the decision making process.
Normalize	To adjust data for effects such as inflation, anomalies, seasonal patterns, technology changes, accounting system changes, and reorganizations.
Parametric Cost Estimating	A cost estimating methodology using statistical relationships between historical costs and other program

	variables such as system physical or performance characteristics, contractor output measures, and manpower loading. An estimating technique that employs one or more cost estimating relationships (CERs) for the measurement of costs associated with the development, manufacture, and/or modification of a specified end item based on its technical, physical, or other characteristics.
Parametric Cost Model	A model is a series of cost estimating relationships (CERs), ground rules, assumptions, relationships, constants, and variables that describe and define the situation or condition being studied. If the model is developed and sold to the public for broad application, it is typically referred to as a commercial model. If the model is developed for the specific needs of an organization, then it is referred to as a proprietary (or company-developed) model.
Platform	Hardware or software architecture of a particular model or family of computers. The term sometimes refers to the hardware and its operating system.
Price Analysis	The process of examining and evaluating a prospective price, without an evaluation of the separate cost elements and proposed profit of the individual offeror whose price is being evaluated. Price analysis may be accomplished by a comparison of submitted quotations; a comparison of price quotations and contract prices with current quotations for the same or similar items; the use of parametric techniques; and/or a comparison of proposed prices with independently developed cost estimates.
Price Negotiation Memorandum (PNM)	A document that accounts for the results of a negotiation. The PNM documents the agreement reached with the successful offeror. It also serves as the permanent record of the decisions the negotiator made in establishing that the price was fair and reasonable.
Production Rate	The number of items produced in a given time period such as a month or a year.
Program Evaluation and Review Technique (PERT)	A management technique designed for the planning and controlling of complex projects. PERT is utilized by constructing a network model of the integrated activities and events in a program or task, and periodically evaluating the time and cost implications as the work progresses. PERT is a management tool requiring an estimate of the expected time to complete each activity. A PERT chart displays the interrelationships among tasks on

	a project, provides information to highlight trouble areas, and allows an evaluator to determine the critical path of events for the project.
Prototyping	An original or model of a final product that is subject to full service testing, and on which a later item is formed or based. Generally, the changes between the prototype model and the production model are those found to be needed during testing.
Real-Time	An immediate response. The term may refer to fast transaction processing systems in business; however, it is normally used to refer to process control applications. For example, in avionics and space flight, real-time computers must respond instantly to signals sent to them. Real-Time may also refer to any electronic operation that is performed in the same time frame as its real-world counterpart. For example, it takes a fast computer to simulate complex, solid models moving on screen at the same rate they move in the real world. Real-time video transmission produces a live broadcast.
Re-engineering	The process of restructuring and/or redesigning an operational (or coded) hardware or software system or process in order to make it meet certain style, structure, or performance standards.
Reuse	A software development technique that allows the design and construction of reusable modules, objects, or units that are stored in a library or database for future use in new applications. Reuse can be applied to any methodology in the construction phase, but is most effective when object oriented design methodologies are used.
Security	A system of safeguards designed to protect computer systems and related data from damage or unauthorized access.
Should Cost Estimate	An estimate of a contract price that reflects reasonably achievable contractor economy and efficiency. It is generally accomplished by a team of procurement, contract administration, cost analysis, audit, and engineering representatives. Its purpose is to identify uneconomical or inefficient practices in a contractor's management and operations by quantifying the findings in terms of their impact on cost, and to develop a reasonable price objective for negotiation.
Software Development Life	The stages and processes through which software passes during its useful life. This includes requirements

Cycle	definition, analysis, design, coding, testing, and maintenance.
Software Tools	Programs that aid in the development of other software programs. Software tools assist the systems/software personnel to perform activities related to designing, coding, compiling, and debugging. CASE tools (previously defined) are an example of software tools.
Technical Evaluator	The study, investigation, or test and evaluation conducted to determine the technical suitability of material, equipment, or a system.
Technical Noncost Data	Non-financial information that describes the item, subsystem or system. Technical noncost data includes scientific or technical information such as engineering documents, engineering drawings, and lines of code.
Top-Down Methods	A method that estimates the overall cost and effort of a proposed project based on the global properties of a past project.
Validation	In terms of parametric estimating, a process used to determine whether the cost estimating relationship and/or model selected for a particular estimate is a reliable predictor of cost for the system being estimated. Validation also includes a demonstration that the model users have sufficient experience and training, calibration processes are thoroughly documented and formal estimating system policies and procedures are established and enforced.
Variable Cost	A cost that changes with the rate of production of goods, production quantity, or the performance of services.
Work Breakdown Structure (WBS)	A WBS is a product-oriented family tree, composed of hardware, software, services, data, and facilities that results from system engineering efforts during the development and production of a material item, and which completely defines the program. A WBS displays and defines the product(s) to be developed or produced and relates the elements of work to be accomplished to each other and to the end product.

References

This appendix is not complete. It will be updated as part of the final production of this handbook.

Chapter 1

Chapter 2

Chapter 3

Gujarati, Domodar, *Basic Econometrics*, McGraw Hill Book Company, New York, 1978.

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Air Force Institute of Technology, Federal Acquisition Institute. June 200. *Contract Pricing Reference Guide, Quantitative Techniques for Contract Pricing, Volume II*.

DCAA Contract Audit Manual, DCAAM 7640.1, January 2007 Edition.

Chapter 5-400, Section 4, *Audit of Information Technology Systems General Internal Controls.*

Chapter 5-1200, Section 12, *Audit of Estimating System Internal Controls*.

Chapter 9-1000, Section 10, *Audit of Parametric Cost Estimates*.

DFARS Section 215.407-5, *Estimating Systems.*

DFARS, Section 252.215-7002, *Cost Estimating System Requirements*.

FAR Part 15, Subpart 15.4, *Contract Pricing.*

FAR Part 15.407-5, *Estimating Systems*.

Public Law 10 U.S.C. 2306a, *Cost or Pricing Data: Truth in Negotiations.*

Chapter 8

Glossary

Boehm, Dr. Barry W., *Software Engineering Economics*, Englewood Cliffs, NJ, Prentice-Hall, 1981.

Capron, H.L., *Systems Analysis and Design*, Menlo Park, CA, Benjamin/Cummings Publishing Company 1986.

Defense Systems Management College, *Cost Estimating Overview*, 1996.

Federal Acquisition Regulation (FAR), 15.401-Definitions, January 1999.

Society of Cost Estimating and Analysis (SCEA), Glossary, 1994.

Software Technology Support Center (STSC), *Report on Project Management and Software Cost Estimating Techniques*, Document Number STSC-TR-012-Apr 95, Hill Air Force Base, UT, 1995.

Webster's Dictionary of Computer Terms, NY, Prentice-Hall, 1988.

Parametric Estimating Handbook User Comments

Please submit any comments regarding the Fourth Edition of the Parametric Estimating Handbook to:

International Society of Parametric Analysts
Attention: Bill Brundick
8156 Blue Ball Rd., Stewartstown, PA 17363
717-993-3501, Fax 717-993-5422
wbrundick@verizon.net

Comments may also be submitted electronically via the International Society of Parametric Analysts (ISPA) web site at www.ispa-cost.org.

When providing comments, suggestions, or specific examples, please reference the chapter number and page number.

Chapter Number:_____________ Page Number:________________

Specific Comments: