Volume 10 | Issue 3 | November 2022

ISSN: 2160-4746

Journal of Cost Analysis and Parametrics

Editor in Chief: Erin K. Barkel, CCEA®

Editor: David L. Peeler, Jr., CCEA®

ICEAA

International Cost Estimating & Analysis Association

www.iceaaonline.com

Journal of
Cost Analysis
and Parametrics

Volume 10 | Issue 3 | November 2022

ISSN: 2160-4746

Editor's Note
David L. Peeler, Jr., CCEA®

Welcome to the third issue of the resurrected Journal of Cost Analysis and Parametrics (JCAP). We thank all who have taken time to write, submit, and revise pieces for publication within these pages. We do not seek to pursue themed editions; but by luck alone, this issue has a good taste of learning curve theory from varying perspectives. Additionally, this issue contains articles that range across the breadth of cost from program management to sensitivity analysis.

The intellectual pursuits revealed by these authors are the results of work papers, school studies, and presentations to our annual workshop. Our first article concludes the summarization of **Danny Polidi**'s doctoral dissertation. In our April 2022 issue, the first installment described generalized block diagram use to create a work breakdown structures. The second article from his dissertation, *A System Engineering Approach Using Sensitivity Analysis for Reducing System Cost*, illustrates the benefits of cost optimization during early program design stages. Applying sensitivity analysis to early design, the highest influences on overall system cost is determined, thus realizing highest performance at minimized cost.

The second article is the first of three dealing with learning curve theory. Captain **Susan L. Moore** was the recipient of the 2022 ICEAA Outstanding Air Force Institute of Technology Thesis Award for *Step-down Functions in Airframe Learning Curves: Do They Exist?* Her research provides an empirical examination as to the existence of step-down functions and whether they should be employed in cost estimates. In the fourth article, *Projecting Future Costs with Improvement Curves: Perils and Pitfalls*, **Brent M. Johnstone** examines common errors in the use of learning curves and the dangers of using them to measure production line efficiency. Finishing out the trio of learning curve articles, **Harry T. Larsen** presents a *Theory of Complex Work* utilizing iterative measurable tasks. He shows that by embedding discrete iterative tasks into a feedback system, the impact on the broader work system can be evaluated.

Article three is our sole software focused piece. In *Are Agile/DevOps Programs Doing Enough Systems Engineering?* Anandi Hira analyzes the difference in Systems Engineering, Integration and Test, and Program Management costs between programs using Agile/DevOps versus a Waterfall approach. **Christina N. Snyder** provides our fifth article, *(CE)^2 – Communication and Empowerment for Cost Estimators: How to Use Soft Skills to Exponentially Impact Your Team's Analysis,* which won the Trending Topics category best paper at the 2022 ICEAA Workshop. She continues her examination of "soft skills" and their application to cost estimating. A topic she also addressed in our April 2022 issue. **Daryl Ono** provides our final article, *A Continuance of Marginal Cost Methodology in Project Change Management*, wherein he discusses change management as an inevitable aspect project engineering critical to economic value.

Happy reading. We hope you enjoy the material and find productive ways to apply it in either your professional efforts or personal interests. Thank you for your continued supported and please keep the manuscripts coming.

David Peeler
JCAP Editor

A System Engineering Approach Using Sensitivity Analysis for Reducing System Cost

Danny Polidi
Mike Crist
Dr. V. Chandrasekar

This document does not contain technology or Technical Data controlled under either the U.S. international Traffic in Arms Regulations or the U.S. Export Administration Regulations

Abstract: Modern software packages exist to estimate system cost early in the systems development and procurement process. The commercially available software which estimates system cost are limited in their ability to aid in system optimization towards multi-objective cost and performance goals, as many require a completed system design. To illustrate the benefits of cost optimization during early stages of design, this paper describes a sensitivity analysis applied to the design of an engineering system. This process seeks to use sensitivity analysis and a spiral design process to determine which cost drivers have the highest influence on overall system cost, and to realize high system performance while minimizing costs. This work is novel in that it describes a method and toolkit to enable simultaneous consideration of system costing with system engineering. This work is novel in that it demonstrates how to determine the cost sensitivities of components in a system, and how the sensitivity values can be used to suggest component parameter variations to maximize the impact to overall system cost. And finally, this work is novel in that it demonstrates a method using component cost sensitivity to determine the range of possible cost improvements to bound project return on investment.

I. Introduction

There are several very good commercially available cost estimation packages. To use these packages, first a system must be defined. The system must be defined in terms of hardware blocks. The hardware blocks can be arranged with a hierarchy such as a work breakdown structure (WBS). Once the system is defined the system can be entered into the cost estimation package. The package essentially converts each hardware component into a corresponding cost. In this way the cost of a system can be determined.

The limitation of the commercially available cost estimation packages is that it is essentially a unidirectional process. A user defines a system and uses the cost estimation package to estimate cost. The user can then experiment with alternatives or modifications to the system and estimate the corresponding associated system costs. What is missing is a bidirectional interaction with the software package. There is very little guidance from the cost estimation package which suggests to the user system modifications for consideration. It lacks suggestions to the designer which modifications would have the greatest impact to the overall cost of the system.

It is desirable to have a feature within a cost estimation package which can analyze the components of the system to determine which components have the greatest impact. In other words, which components have the highest sensitivity for modification as it pertains to the overall cost of the system.

Although sensitivity analysis is well understood the application of sensitivity analysis upon a cost model for the purposes of maximizing the impact to the overall system cost is novel. This paper explores an effort to generate a cost sensitivity algorithm of the various components in a system to analyze a system and determine which subsystem components in a chosen design solution have the highest sensitivity to cost for the overall system. In addition, the analysis highlights the areas to which a system designer could apply focus to reduce the overall system cost early in the life cycle of a Program.

The focus of this paper will be divided into five main topics: 1) an understanding of the current industry capabilities, 2) development of a cost sensitivity algorithm for application upon a system cost model including the development of sensitivity key size metrics (KSMs) for the component parameters for use with the cost sensitivity algorithm, 3) an example application of the full cost sensitivity algorithm with KSMs on a sample system cost model, 4) an example application of the full cost sensitivity algorithm with KSMs on a "real" system cost model, and 5) a brief discussion on return on investment (ROI) utilizing the results of the cost sensitivity algorithm.

In the first section, the tools available in industry are explained. Specifically, how cost packages are structured and designed for use. It will be explained that the available tools offer a user the ability to take a specific system architecture and estimate an expected cost for that system. There does not currently exist a tool which can offer a robust examination of the system and offer feedback to the user on how to improve or optimize the system architecture. It is that lack of feedback in the process which is addressed within this paper. Specifically, the use of a cost sensitivity algorithm to highlight areas of focus which can most significantly impact overall system cost.

In the second section, the development of a sensitivity algorithm is presented. It will be demonstrated that each parameter within a component could be considered either minor or impactful to overall system cost. The impactful parameter could then be varied (up or down) by some arbitrary amount to affect overall system cost (up or down). It is shown that the variation of the impactful parameter drives the estimated cost away from the baseline cost differently depending upon the component for which it applies. These differences constitute a cost sensitivity of a component parameter upon the overall system cost. These cost sensitivities can then be collected for each parameter and analyzed to determine the relative ranking of the cost sensitivity parameters. The highest-ranking cost sensitivity parameters are of particular importance to a system designer interested in optimizing a system architecture for cost versus performance trade studies.

A limitation was identified in the development of the cost sensitivity algorithm related to the factor by which the component parameters were varied to calculate cost sensitivity. Included is the discussion devoted to resolving the limitation of the usage of an arbitrary and uniform variation factor. Instead, KSMs were determined which allow for unique variation factors for each type of parameter.

In the third section, the fully developed cost sensitivity algorithm with KSMs was applied to an example cost model. The result of applying the algorithm is presented and demonstrates how the results can be applied to show potential cost improvements.

In the fourth section, the fully developed cost sensitivity algorithm with KSMs was applied to a "real" example cost model based on a generalized work breakdown structure (WBS) for a RADAR system applied to military applications in the aerospace industry. The result of applying the algorithm is presented and demonstrates the significant impact achievable to overall system cost when focus is applied appropriately to the areas for which overall system cost is most sensitive.

In the fifth section, the algorithm highlights the areas where trade studies could be performed and yields a target return on investment (ROI) budget, or a limit of money to spend as investment to achieve those cost improvements.

This work is novel in that it describes a method and toolkit to enable simultaneous consideration of system costing with system engineering. This study demonstrates how to determine the cost sensitivities of components in a system, and how the sensitivity values can be used to suggest component parameter variations to maximize the impact to overall system cost. And finally, this work is novel in that it demonstrates a method using component cost sensitivity to determine the range of possible cost improvements to bound project return on investment.

II. Related Work

One of the issues which complicates predicting the cost for any system is the concept of size. Systems which might have the same general function could vary significantly in terms of cost. For example, the engine in a supercar is more expensive than the engine in a commuter car even though they are both engines and have the same general function. It then becomes an exercise to understand what about those systems yield such significantly diverse costs. This concept is referred to as size. The application of sizing is not limited to a system. For example, it could be applied to a financial institution [1]. The referenced author recognizes a base cost and then on top of that are what the author calls "anomalies." The author writes "The purpose of normalization is to eliminate such anomalies and provide accurate historical information that enables reliable comparisons and forecasting." Normalization could be applied to variations in agricultural costs to normalize prices [2]. Again, this is the concept of a size with anomalies due to seasonal variations which must be normalized to determine standardized pricing. In a more generalized approach to normalizing cost data, the data could first be divided into broad categories. For example, the data could be grouped into three categories: normalizing for content, normalizing for quantity, and normalizing for inflation [3]. But again, the concept is the same. There is a base normalized size with factors which complicate the data. The International Cost Estimation and Analysis Association (ICEAA) offers instructional courses in cost estimation. The module "Data Collection and Normalization" [4] is devoted to this topic. Here cost can be normalized in one of three categories: Cost Units, Quantities, and Sizing Units. And for Sizing Units, the subcategories include weight, density or volume, and for software, Lines Of Code (LOC).

The variables that comprise a system may also influence the cost of the system. These variables are referred to as cost drivers. "A cost driver is the direct cause of a cost and its effect is on the total cost incurred" [5]. Or more appropriately, "Cost drivers are the structural determinants of the cost of an activity, reflecting any linkages or interrelationships that affect it" [6]. Of course, these definitions do not make a distinction between cost drivers which are impactful verses non-impactful. While any variables can contribute towards cost, each will have different sensitivities associated with them.

Related to the topic of cost drivers to a system is the concept of a trade study. A trade study is defined as "the activity of a multidisciplinary team to identify the most balanced technical solutions among a set of proposed viable solutions" [7]. This is a generalized definition but its applicability toward cost drivers applies. A trade study can be further defined where a "trade study is a formal tool that supports decision making" [8]. And in this reference, it notably applies to "realistic alternatives" and includes objects such as "performance" and "cost."

However, the reference fails to adequately offer a solution as to how a cost trade study may be conducted. A system trade study can be performed using a standardized approach [9] based on the more generalized 'Standard Approach to Trade Studies' [10]. This more focused approach offers "significant developments" in cost benefits resulting from the trade study. The referenced author offers Cost As an Independent Variable (CAIV) and draws a connection between system cost and the results of the system trade.

However, in this reference the topic of "cost" and "risk" are deliberately removed from the "tradable criteria list". Whereas for a typical trade study, cost and risk are two of the main criteria for decision making. The reference is a typical approach for bidding where the system performance is defined early, and cost is included towards the end of the evaluation. To "converge on recommendations that are robust in the presence of uncertainty" a framework for a standardized trade study may be employed [11]. However, although the article does devote much attention to the area of cost, it does not offer any insight into cost as a variable with which to optimize a solution.

When considering many variables, or categories of variables, the topic of Multi Criteria Decision Analysis (MCDA) appears relevant. MCDA is used "as an umbrella term to describe a collection of formal approaches which seek to take explicit account of multiple criteria in helping individuals or groups explore decisions that matter" [12]. The author describes a structured approach to decision making which serves to help "decision makers to understand and to define their preferences, rather than descriptive, describing what they do and seeking simply to elicit their preferences." However, the article does not include a use case where discreet variables rather than stakeholders are considered.

The same observation is applicable for several works regarding MCDA. Considerations such as "economic, social, and environmental criteria are nowadays involved in practically all decision situations" [13]. The referenced author indicates the "decision process should naturally explore the conflicting nature of the criteria, the corresponding tradeoffs, the goals set by the decision makers, and of course the way that these can be introduced in an appropriate decision model that takes into account the subjectivity of the decision process and the preferences of the decision makers." The author describes "discrete problems." But by "discrete problems" the author refers "to decision situations involving the evaluation of a finite set of alternatives and actions over a predefined set of evaluation criteria." To illustrate, the author considers "a company or a public institution, where a manager and/or a group of people are confronted with a decision situation or "problem" that requires them to make a decision." Although the article is devoted to decision making it approaches decision making from a more global view with a wide variety of contributing factors and does not offer any specific insight into the cost variable.

A more refined approach to decision making is developed in the Analytic Hierarchy Process (AHP) and the Analytic Network Process (ANP). Both processes have criteria and each criteria have a value and a weight. By combining value and weight for each criteria a net result can be obtained which indicates the preferable decision choice. In the case of AHP [14] the decision is structured into a hierarchy with a goal, decision criteria, and alternatives. The Analytic Network Process (ANP) [15] is a more general form of AHP where the decision is structured as a network. For multicriteria analysis including AHP & ANP each criteria is associated with a weight of importance. Importance in this context is a relative measure between various criteria. By contrast each criteria may also have a degree of criticality. By critical, we mean the degree to which a change in that criteria's weight affects the final decision. It is possible that a criteria with a small weight in importance may be more critical to the final decision. This is the concept of sensitivity.

Although the topic of sensitivity analysis has been explored, the application of sensitivity analysis upon a system's cost is limited. In "An Introduction to Sensitivity Analysis" [16] the referenced author offers an introduction to sensitivity analysis using a series of papers on the subject. The article relies on the STELLA software which is an application for system modeling. The author writes that "Sensitivity analysis is used to determine how "sensitive" a model is to changes in the value of the parameters of the model and to changes in the structure of the model." The article describes exploratory exercises where the function of the system is described, and the sensitivities of the various system elements are considered. However, the techniques described do not mention cost as a variable. An alternative approach where system sensitivities are reduced to limit the effect upon variations in process parameters may be employed [17]. This varies from the concept of normalization in that normalization removes variations such that meaningful comparisons could be made. Whereas in this reference the variations are removed to dampen the effects for enhanced performance as in the case of a control system. A "process of recalculating outcomes under alternative assumptions to determine the impact of a variable under sensitivity analysis can be useful for a range of purposes" [18]. The reference specifically mentions both an "increased understanding of the relationships between input and output variables in a system or model" as well as "enhancing communication from modelers to decision makers." These purposes could be applied upon a system to determine cost sensitivities of the various elements to allow a system designer to understand the relationships and make informed decisions.

An interesting study of a production inventory system made a case for the application of sensitivity analysis to bring the "model solutions closer to the complexities of real systems" [19]. The referenced author makes the case that in the absence of sensitivity analysis the designer's predictions rely on history and assumes the same trend. While this may be a good assumption, by using sensitivity analysis it becomes a more predictive method, not solely based on history but with some understanding of the sensitivity of the variables. Cost was a secondary consideration and limited by first understanding the inventory levels and then calculating the corresponding costs. A more impactful result might demonstrate, for example, that cost could be significantly reduced if production runs were modified in a quantifiable way, and therefore inventory would have to be modified accordingly.

Sensitivity analysis can be applied in "many fields such as environmental risk assessment, behavior of agronomic systems, structural reliability or operational safety" [20]. The author explains that even "an environmental impact problem may be framed through the lenses of economics, and presented as a cost benefit or risk analysis, while the issue has little to do with costs or benefits or risks and a lot to do with profits, controls, and norms." The referenced author mentions sensitivity analysis "provides users of mathematical and simulation models with tools to appreciate the dependency of the model output from model input, and to investigate how important is each model input in determining its output." Although the referenced author explains sensitivity analysis can be used to investigate the dependency of output to input variables there is no significant example demonstrating a dependency with regards to cost.

The study on the topic of ownership of an electric transportation system in Swedish medium sized cities aimed "to emphasize on sensitivity analysis for the total cost of ownership (TCO) to reduce uncertainty by identifying which factors of interest that most likely cause the estimated cost values for the electric bus" [21]. The study does help to illustrate how versatile is sensitivity analysis and that it can be used to address cost in a wide variety of applications. However, the study focused on infrastructure as a system where the term "system" is very broad in its application. The study is diminished in that it starts with the assumption that electric alternatives reduce greenhouse gases but neglects to address the contributions of pollution created in the generation of electric power. The study upon marine renewable energy uses sensitivity analysis for cost reduction [22]. The analysis which largely focuses on cost "highlights the sensitivity of marine energy to three key parameters: the capital cost of first devices, the level of deployment before sustained cost reduction emerges, and the average rate of cost reduction with deployment (learning rate)." In this case the analysis focuses more on the different phases in the lifecycle of the system rather than on the elements of a system. While both include cost and sensitivity neither reference considers a system as a collection of components as in the case of a RADAR system.

There are numerous available Commercial Off The Shelf (COTS) cost modeling tools. In addition to providing the cost for an existing design, the PRICE Cost Analytics tool offers the system designer an ability to translate "needs" into "requirements" [23]. As noted for Design-to-Cost Targets, "It is widely accepted that 80% to 90% of cost is determined at the design or development stage." This highlights the need to perform trade studies early to optimize a solution before the design architecture has been defined. The Constructive Systems Engineering Cost Model (COSYSMO) developed at MIT is an industry standard "to estimate the Systems Engineering effort for large-scale systems (both software and hardware)" [24]. Unfortunately, the tool focuses on the cost associated with the system engineering aspects of the design rather than a more comprehensive estimation of all disciplines associated with development or production costs. SEER by Galorath offers a tool for a system engineer to estimate a system cost once the system has been defined [25]. And the tool offers some rudimentary features regarding sensitivity. By varying the parameter inputs to the cost model, the user can observe the effect on overall system cost. But to utilize this ability to determine cost sensitivities for every parameter is manual and labor intensive. What is missing from all these COTS packages is the ability to directly calculate a cost sensitivity of each component in the system design to direct a system designer towards a cost optimized solution. This is not to imply that individual system component costs directly translate into the total cost of the system. Alternative selections of components would likely require some wider consideration of components which would necessarily have some cost increases and decreases. Instead, this method is a practical solution for a real problem. This work is intended to provide a designer a "compass" on where to focus attention by identifying the components with the highest cost sensitivities. This work is novel in that it describes a method and toolkit to enable simultaneous consideration of system costing with system engineering. This work is novel in that it demonstrates how to determine the cost sensitivities of components in a system, and how the sensitivity values can be used to suggest component parameter variations to maximize the impact to overall system cost. And finally, this work is novel in that it demonstrates a method using component cost sensitivity to determine the range of possible cost improvements to bound project return on investment.

III. Sensitivity Algorithm Development

Contrary to other efforts where cost has been normalized out of the equation or removed altogether this paper directs the effort toward the beginning of the design life cycle to optimize a solution before the final architecture has been selected.

A sensitivity algorithm was applied in several steps. First, a sample system cost model was identified such that the algorithm could be tested upon that sample. Next, the uncertainty parameters were determined. Then, the range of variation was determined. And finally, the results were calculated.

A. Selection of a System Cost Model

The intent here is not to create a cost estimation tool. Instead, what is presented is a practical solution for a real problem used in conjunction with a COTS tool. It is assumed the math behind the analysis performed by the COTS tool forms a sufficient foundation upon which to develop a sensitivity algorithm. The outcome of this work is intended to provide a designer a “compass” on where to focus attention by identifying the components with the highest cost sensitivities. For purposes of creating a sensitivity algorithm for use with a cost estimation tool it is necessary to have a cost model upon which to apply an algorithm. There does exist a detailed and generic WBS structure which was developed for a RADAR system applied to military applications in the aerospace industry [26]. However, in the early stages of algorithm development it is sufficient to use a sample cost model. Most commercially available cost estimation packages come with a library of sample cost models. These sample cost models were surveyed to find an example which was complex enough to contain a significantly large WBS structure to allow for analysis while at the same time was not so large as to inhibit the process of algorithm development.

A cost model provided by SEER Galorath was identified and can be seen in Table 1. The WBS is indented down to four levels. The WBS includes both analog and digital subsystem blocks (e.g., Receiver Module, Digital Processing) as well as structural components (e.g., Receiver Chassis). This sample cost model was determined to provide for a significantly large enough WBS structure to allow for analysis. Also, the quantity of components, in this case nineteen, should

WBS Number	Component
1	NewGen Listening Station
1.1	Equipment Configuration
1.1.1	Receiver Module
1.1.1.1	Receiver
1.1.1.2	RF Module
1.1.1.3	RF Machined Housing
1.1.1.4	Receiver Chassis
1.1.2	Digital Processing
1.1.2.1	Converter & Noise Reduction
1.1.2.2	Data Processing
1.1.2.3	Purchased Memory
1.1.2.4	Interconnect – Data Bus
1.1.2.5	Instrumentation Panel
1.1.2.6	Digital Processing Chassis
1.1.2.7	Controller Software
1.1.3	Misc. Equipment
1.1.3.1	Wire Interconnects
1.1.3.2	Purchased Racks
1.1.3.3	Purchased Power Supply
1.2	Operational and Support Sites
1.2.1	Northeast Auxiliary
1.2.2	Atlantic Operations Center
1.2.3	Western Operations Center
1.2.4	Midwest Repairs
1.2.5	Express Repairs

Table 1: Sample Cost Model WBS Structure

provide for a significant quantity of component parameters with which to experiment.

B. Definition of Terms

To standardize terminology some definitions are presented. The definitions are for the current paper and all efforts were made to adhere to conventional industry definitions. The following sections will elaborate considerably and provide context.

1. **System** – The System refers to the highest level WBS item.

2. **Subsystem block** – The Subsystem block refers to the highest-level hardware within the WBS structure below the System. As in Table 1, examples are 1.1.1. Receiver Module or 1.1.2. Digital Processing.

3. **Component** – The Component refers to the lowest level hardware within the WBS structure. As in Table 1, examples are 1.1.1.1. Receiver, 1.1.1.2. RF Module, etc.

4. **Parameter** – A Parameter in this context is a variable associated with a Component. An example may be Total Printed Circuit Boards (n) or Circuit Board Size (in2).

5. **Impactful Parameter** – In the more general usage a key cost driver impacts more significantly overall cost than other factors. In this context it was necessary to find Parameters which more so than others affect the overall System cost. Although a Component may have many Parameters, only a small subset of those could be considered an Impactful Parameter based on its effect upon cost.

6. **Variation Factor** – When an Impactful Parameter value is changed by a fixed percentage, the percentage by which it is changed is the Variation Factor.

C. Identification of Minor vs. Impactful Parameters

Sensitivity analysis is used to determine the relationships between independent variables and dependent variables under certain conditions [27]. In this case, it is desirable to determine the effect of changes to parameter values (input) on the overall cost of a system (output).

The sensitivity analysis method consists of three steps. First, the uncertainty parameters are determined. Second, the range of variation is determined. And third, the results are calculated [28]. In this case, the uncertainty parameters are the parameters included in the cost model, the range of the parameters will be a fixed percentage variation which is initially arbitrarily assigned, and then the effect of modifying the parameters will be observed as the output of the cost model. The development of the process is described in detail along with the results.

The first step for the sensitivity analysis is to determine the model parameters which apply for this sensitivity analysis. In general, the parameters for a component are all the same if the components are similar. An analog amplifier and an analog filter may be similar and have the same parameters while a chassis would have very dissimilar parameters.

It is understood that all the parameters of a component contribute to cost in some way. However, not all parameters contribute equally. In some cases, the impact to overall system cost may be quite negligible. Prior to automating the process, it can be quite prohibitive to utilize every parameter indiscriminately. However, if the set of parameters could be limited then achieving meaningful results with a manual method becomes practical. It therefore becomes necessary to determine which parameters could be considered Minor Parameters versus Impactful Parameters.

The available cost parameters were screened for their effect on overall system cost. The top six

EE Impactful Parameters	ME Impactful Parameters
Total CCA	Weight
PCB Size	Volume
Discreet Components per PCB	
Integrated Components per PCB	
Clock Speed	

Table 2: Parameter Survey Results

contributors in the case of an electrical component and the top three contributors in the case of a mechanical component were selected. These selected parameters were identified as Impactful Parameters as opposed to Minor Parameters.

The entries in Table 2 indicate the results of the parameter survey. The parameters fall under either one of two categories: Electrical Impactful Parameters or Mechanical Impactful Parameters.

The second step in a sensitivity analysis is to determine the range by which the parameters are varied. In this case, instead of assigning a range of values, an initial fixed variation factor was selected to yield a sufficient spread in the results to demonstrate the method and begin to draw some conclusions. An initial fixed variation factor was arbitrarily assigned as 20%. A table was created which shows all the components with their respective parameters and their variation factors (Table 15 column F).

D. Variation of a Single Parameter

For the third step of sensitivity analysis the results are calculated. And in this case, the modification of the cost model parameters will be observed and recorded to understand the effect on the output of the overall cost of the system. This third step is where much of the work occurs. The development of the process is described in detail along with the results. The process demonstrates the method first with a single parameter before then demonstrating an example with all Impactful Parameters.

1) ESTABLISHING BASELINE COST

To understand the effects of modifications to the cost model to overall system cost it becomes necessary to establish a baseline, or a baseline cost. In this case, a sample cost model was chosen from the library of existing cost models from the commercially available package.

The selected model had the WBS structure as it appears in Table 1. It can be seen there is a system, subsystem blocks and components. After the specific model was selected, the application was run to estimate the cost for the overall system.

For purposes of illustration the following discussion will be applied to a specific parameter within a specific component. In this case, the number of PCBs in the Receiver will be analyzed. This case appears in Table 15, row 5.

2) VARY COMPONENT PARAMETER UP 20% FOR "UP" COST

The selected parameter is varied up by an amount of 20%. For example, the number of CCAs in the component is modified from 2 to 2.4 (Table 15, row 5, column G). It is understood that practically it is unrealizable to have 2.4 CCAs and that only whole integers are possible. However, the values are strictly theoretical and used to determine the sensitivity of a particular parameter. After the sensitivity factors have been determined the paper will suggest the selection of practical and realizable values, e.g. 1 vs 2 CCAs. The new parameter value is applied to the cost model and the modified overall system varied "up" cost is estimated (Table 15, row 5, column K). The varied "up" cost is then compared to the baseline cost to estimate a delta "up" cost value (Table 15, row 5, column N).

3) VARY COMPONENT PARAMETER DOWN 20% FOR "DOWN" COST

The selected parameter is then varied down by an amount of 20%. For example, the number of CCAs in the component is modified from 2 to 1.6 (Table 15, row 5, column H). The new parameter value is applied to the cost model and the modified overall system varied "down" cost is estimated (Table 15, row 5, column L). The varied "down" cost is then compared to the baseline cost to estimate a delta "down" cost value (Table 15, row 5, column O).

4) CALCULATE THE AVERAGE DELTA AND RANGE

Referring to Figure 1A, the varied "up" and varied "down" costs can be seen graphically with respect to the baseline cost. When a parameter is varied from its baseline value it has the effect of driving the overall system cost away from the baseline. To put the data in a useful format the absolute value of the two results is graphed, see Figure 1B. In this figure both deviations are illustrated as driving the cost positively away from the baseline cost. And it can be observed that the degree by which the two deviations drive the cost away from the baseline is not the same. It should be noted that if the parameter resides upon a linear portion of a cost curve, then these two deltas would be identical. The fact that they are not the same indicates there is some sort of non-linearity for the cost curve. For purposes of this analysis, it is unnecessary to fully understand the cost curve. It is the magnitude of each delta which is of particular importance to the current discussion.

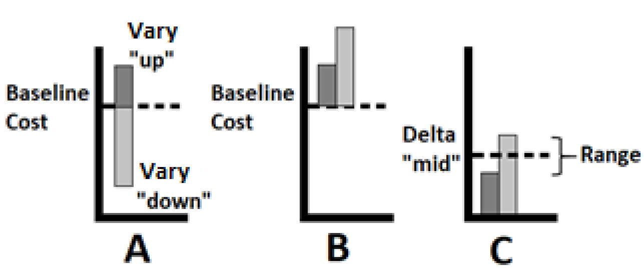


Figure 1. Variation of Cost.

The two deltas are then normalized and averaged. The result is the delta "mid" value and the "up" and "down" deltas form the range, see Figure 1C. This gives a quantitative value for the sensitivity of the one parameter for a component upon the overall cost of the system. For this specific example, the numerical equivalent of Figure 1 appears in context in Table 15, row 5, columns K – O.

E. Expanded Analysis for Every Parameter

The ability to associate cost sensitivity to the various components of a system was previously, but briefly explored in an Excel based cost model [29]. The current paper parallels to a small degree the Excel effort where the cost model, which after cost calculations, had the ability to evaluate the sensitivity of various architectures associated with the design. The results were color coded by impact such that the user could identify where to apply focus to have the greatest effect on cost.

With the sensitivity algorithm established for a single parameter the next step is to apply the algorithm to every Impactful Parameter in the cost model and calculate all cost sensitivities. The full table of values appears in Table 15. The set of Impactful Parameters was chosen and appear in Table 15, column D. The varied amount, as discussed, was a uniform value of 20% (Table 15, column F). The spreadsheet calculates the varied "up" and varied "down" parameter values (Table 15, columns G & H). The cost tool was then run repeatedly and for each consecutive run only one parameter from the list was changed keeping all other values in their baseline condition. The overall system cost was then collected for each permutation of parameter value (Table 15, columns K & L). Using the results of each run from the cost tool the delta "mid" and delta "range" values were calculated for each parameter (Table 15, columns P & Q). With the delta "mid" values calculated the values could be ranked in order of greatest to least impact to overall system cost. The ranking appears in the

Parameter	Component	Sensitivity or Delta "mid"	Rank Of Impact
Clock Speed (USD/MHz)	Converter & Noise Reduction	$11.29K	14
	Data Processing	$11.15K	15
	Interconnect – Data Bus	$3.66K	21
	Purchased Memory	$115	32
	Receiver	$8.18K	16
	RF Module	$7.16K	17
Discreet Comp per PCB (USD/n)	Converter & Noise Reduction	$4.37K	20
	Data Processing	$2.60K	22
	Interconnect – Data Bus	$5.90K	19
	Purchased Memory	$237	31
	Receiver	$2.36K	23
	RF Module	$555	27
Integrated Comp per PCB (USD/n)	Converter & Noise Reduction	$147.20K	4
	Data Processing	$111.79K	5
	Interconnect – Data Bus	$31.83K	9
	Purchased Memory	$1.53K	26
	Receiver	$99.83K	6
	RF Module	$17.32K	13
Volume (USD/ft3)	Digital Processing Chassis	$5	36
	Instrumentation Panel	$0.09	38
	RCV Chassis	$4	37
	RF Machined Housing	$53	34
Weight (USD/lb.)	Digital Processing Chassis	$22.36K	10
	Instrumentation Panel	$7.08K	18
	RCV Chassis	$19.17K	11
	RF Machined Housing	$18.41K	12
Total CCAs (USD/n)	Converter & Noise Reduction	$283.92K	2
	Data Processing	$351.13K	1
	Interconnect – Data Bus	$50.92K	8
	Purchased Memory	$1.68K	24
	Receiver	$171.68K	3
	RF Module	$82.07K	7
PCB Size (USD/in2)	Converter & Noise Reduction	$441	28
	Data Processing	$380	29
	Interconnect – Data Bus	$61	33
	Purchased Memory	$13	35
	Receiver	$265	30
	RF Module	$1.56K	25

Table 3. Initial Sorted by Parameter

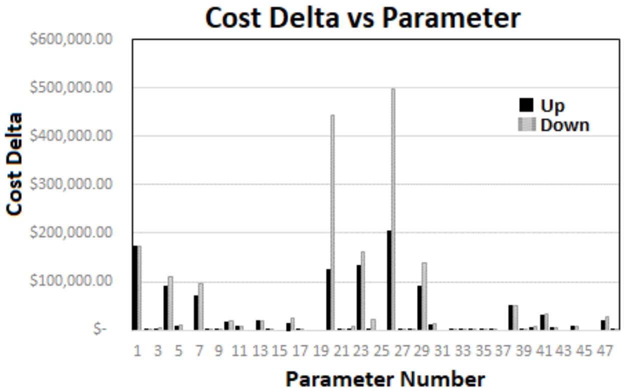


Figure 2. Cost Delta vs. Parameter

table and is color coded with the ten most Impactful Parameters as Red, the next ten as Yellow, the next ten as Green, and the remainder as uncolored (Table 15, column R).

F. Interim Results - Uniform Variation Factor

The results are graphically illustrated in Figure 2. The various parameters have a wide variety of impact to overall system cost. This is as expected.

In addition, the results were sorted by parameter, see Table 3. It is very clear from Table 3 that for any parameter there exists a set of components upon which the parameter applies. And for each of the components there is clearly a difference in the sensitivity of that parameter depending upon to which component it is applied. In the case of Clock Speed, for example, the parameter is associated with six different components. The Converter and Noise Reduction component has the highest sensitivity for this parameter, and in addition, ranks as 14th most Impactful Parameter in sensitivity for the entire system.

In Table 4 the maximums for each parameter are collected. In other words, for the parameter of Total CCAs it was determined in Table 3 that of the six components, Data Processing had the highest sensitivity and is in fact ranked as 1st overall. Therefore, in Table 4 for the parameter of Total CCAs, only the component Data Processing is listed with its corresponding sensitivity or delta "mid" value. The same logic applies for all the other parameters listed in Table 4.

In addition, all cost scenarios were numbered in order of overall impact to system cost (Table 15, column R). Included are the corresponding components along with the parameter which influences the component costs. As mentioned, the ranking was identified grouped with red for the highest impact or sensitivity (1-10), yellow for medium (11-20) and green for low (21-30). It can be seen, for example, that the total number of CCAs in the Data Processing component has the highest cost sensitivity and was ranked correspondingly with a value of 1.

The results of the analysis are beginning to demonstrate some real-world implications. As a

Parameter	Variation Factor	Component	Sensitivity or Delta "mid"
Total CCA	20%	Data Processing	$351.13K USD/n
Integrated Components per PCB	20%	Converter & Noise Reduction	$147.20K USD/n
Weight	20%	Digital Processing Chassis	$22.36K USD/lb.
Clock Speed	20%	Converter & Noise Reduction	$11.29K USD/MHz
Discreet Components per PCB	20%	Interconnect – Data Bus	$5.90K USD/n
PCB Size	20%	RF Module	$1.56K USD/in2
Volume	20%	RF Machined Housing	$189 USD/ ft3

Table 4. Parameter Maximums

Figure 12

system designer this information is very useful. It indicates to a system designer which piece of hardware should receive focus to reduce cost to the overall system. In other words, for this system, the system designer should consider reducing the number of CCAs in the Data Processing component or increasing the number of Integrated Components per PCB in the Converter & Noise Reduction component, etc.

What is interesting to notice in Table 4 is that for two separate parameters (Integrated Components per PCB and Clock Speed), the Converter & Noise Reduction component had the biggest impact. This implies that if a system designer can only focus resources on one component, that component should be the Converter & Noise Reduction component since it clearly has the potential, with some improvements or modifications, to have the greatest impact to overall system cost. In practice simply swapping out a block and estimating cost gives a first order indication of the cost impact. It is not possible to swap out cost as modular blocks and estimate new costs without understanding that there are affects to the system. As design choices are made, impacts are assessed, costs can be estimated, new choices are made, and eventually the design spirals into a solution. As mentioned previously, this technique is intended as a tool for a cost analyst to provide a designer a "compass" on where to focus attention by identifying the components with the highest cost sensitivities.

G. Key Size Metric (KSM) Development

The data has been sorted by parameter, see Table 3. The first grouping represents all the occurrences where the numerical parameter for the Clock Speed was adjusted or varied. Varying the parameter had a different impact to overall cost depending upon which component contained that parameter. As can be seen in the table the component which had the biggest impact to overall cost when varying the parameter Clock Speed was the Converter & Noise Reduction component.

One significant issue to be addressed is that a uniform variation factor of 20% was used for every parameter. With some consideration it seems that using a uniform variation factor for every parameter is not sufficient and may yield misleading results. For example, consider the decision to vary total number of CCAs by the same factor as Clock Speed. In the case of the number of CCAs it may be reasonable, for example, to reduce the design from 3 to 2 CCAs while at the same time it may be possible to double the Clock Speed. Clearly it is not computationally sensible to consider the unit step size to be the same from one parameter to the next.

To overcome this limitation a set of Key Size Metrics (KSMs) must be developed. The KSMs would specify a unique value (other than a uniform 20%) for each parameter. In this way the relative impact to overall cost between parameters could be determined.

The first step is to establish a sorting of the parameters in order of impact. To do this some amount of engineering judgement and some familiarity with real systems and the associated cost is required. As an example, consider the parameter Discreet Components per PCB. With

Expected Parameter Sequence	KSM To Yield The Expected Sequence
Total CCA	20%
Weight	60%
Integrated Components per PCB	5%
Clock Speed	20%
PCB Size	40%
Discreet Components per PCB	10%
Volume	75%

Table 5. Key Size Metric (KSM) Values

some effort it may be possible to combine various discreet parts together and in so doing reduce the quantity. This of course would have some impact on overall system cost. If enough of these improvements could be achieved, then it may be possible to reduce the size of the PCB. In this way it can be considered that the parameters should be arranged in a hierarchy of impact.

In addition, consider the possibility to reduce the count of CCAs. With some effort it may be possible to reduce the count of CCAs. But in a practical sense it does not ordinarily occur where a majority of the CCAs could be eliminated. Instead, it is a slight reduction in count. However, the impact of that reduction is typically very significant. By comparison, altering the Clock Speed may be significant in value (doubling the clock speed, for example) and may have an impact to cost as well. Typically, the impact is considerably less than reducing the CCA count. With considerations such as these in mind the parameters were sorted, and the results appear in order in Table 5.

The set of maximum parameters listed in Table 4 were used. These entries represent the cases for the components with the greatest impact to overall system cost. The goal was to adjust the variation factor for these few cases such that the resulting sequence of impact would match that as indicated in Table 5. Observing the effect on the overall cost impact the variation factor was varied for each of these parameters. Eventually, through a rigorous method of trial, error, and extrapolation, KSM values indicated in Table 5 were determined.

Using these KSM values for every parameter within the cost model the revised estimated overall system cost should yield results in this same sequence. The next step was to repeat the analysis of the previous sections, obtain another full set of data and analyze the results. The analysis should yield results in the same sequence as that indicated in Table 5.

IV. Example Sensitivity Algorithm Applied to Sample Cost Model

With the sensitivity algorithm established and with a suitable set of KSMs derived to vary each parameter with a unique value the previous effort was repeated.

The same system cost model was used as in the previous effort. The full table of values appears in Table 16. Because of the same subsystem blocks, the same list of applicable Impactful Parameters was chosen (Table 16, column D). The baseline values for each parameter remain unchanged (Table 16, column E). The new KSMs are applied (Table 16, column F). As before the spread sheet calculates the varied "up" and varied "down" values for the parameters (Table 16, columns G &

Parameter	Variation Factor	Component	Sensitivity
Total CCA	20%	Data Processing	$351.13K USD/n
Weight	60%	Digital Processing Chassis	$67.94K USD/lb
Integrated Components per PCB	5%	Receiver	$38.86K USD/n
Clock Speed	20%	Converter & Noise Reduction	$11.29K USD/MHz
PCB Size	40%	RF Module	$5.07K USD/in2
Discreet Components per PCB	10%	Interconnect – Data Bus	$1.99K USD/n
Volume	75%	RF Machined Housing	$189 USD/ft3

Table 6. Parameter Maximums

H). The cost tool was then run repeatedly and for each consecutive run only one parameter from the list was changed keeping all other values in their baseline condition. The overall system cost was then collected for each permutation of parameter value (Table 16, columns K & L). Using the results of each run from the cost tool the delta "mid" and delta "range" values were calculated for each parameter (Table 16, columns P & Q). With the delta "mid" values calculated the values could be ranked in order of greatest to least impact to overall system cost. The ranking appears in the table and is color coded, as before, with the ten most Impactful Parameters as red, the next ten as yellow, the next ten as green, and the remainder as uncolored (Table 16, column R).

A. Results--Nonuniform Variation Factor (KSM)

The results were collected similarly as were done in Figure 2. Consistent with the earlier example the various parameters have a wide variety of impact to overall system cost.

The results were numbered by parameter (Table 16, column R). It is very clear from Table 16 that for any parameter there exists a set of components upon which the parameter applies. And for each of the components there is clearly a difference in the Sensitivity of that parameter depending upon to which component it is applied. In the case of Clock Speed, for example, the parameter is associated with six different components. The Converter and Noise Reduction component has the highest sensitivity for this parameter, and in addition, ranks as 15th most Impactful Parameter in sensitivity for the entire system.

In Table 6 the maximums for each parameter are collected. In other words, for the parameter of Total CCAs it was determined in Table 16 that of the six components, Data Processing had the highest sensitivity and is in fact ranked as 1st overall. Therefore, in Table 6 for the parameter of Total CCAs only the component Data Processing is listed with its Corresponding sensitivity, or delta "mid" value. The same logic applies for all the other parameters listed in Table 6.

In addition, all cost scenarios were numbered in order of overall impact to system cost (Table 16, column R). Included are the corresponding components along with the parameter which influences the component costs. As mentioned, the ranking was color coded with red for the highest impact or sensitivity (1-10), yellow for medium (11-20) and green for low (21-30). It can be seen, for example, that the total number of CCAs in the Data Processing component has the highest cost sensitivity and was ranked correspondingly with a value of 1.

V. Return on Investment (ROI) Sample Cost Model

While all the calculations and results presented thus far are of theoretical importance the value of this work lies in the application of the results. The question of primary concern relates to how a system architecture can be optimized in terms of performance and cost early in the life cycle of a program. To illustrate the significance of the

Component	Parameter	Rank Of Impact	Parameter Baseline	Parameter Try
Data Processing	Total CCAs	1	2	1
Converter & Noise Reduction	Total CCAs	2	3	2
Receiver	Total CCAs	3	2	1
RF Module	Total CCAs	4	0.5	0.5
Digital Processing Chassis	Weight	5	18	17
Rcv Chassis	Weight	6	15	14

Table 7. Baseline/Try Parameter Value

results upon potential improvements to the sample system the following discussion is offered.

The information which appears Table 16, column R indicates the top 30 most Impactful Parameters. It is unrealistic to consider improvements to a system in such a broad number of parameters over a broad set of components. Instead, focus will be applied to a more conservative subset. For consideration, the top five most Impactful Parameters will be analyzed such that the potential improvements to the sample system can be determined. Table 7 lists the top six most Impactful Parameters.

The next step was to assign Baseline/Try values, Table 7. The column Parameter Baseline indicates the value of the parameter which was used in the baseline cost estimations. Observing Table 7, a system designer should understand that the total number of CCAs in the Data Processing component is the most sensitive parameter within the entire system and has the greatest impact to overall cost. Therefore, a system designer should focus resources at this location to optimize the system for performance vs. overall system cost. With some effort, as an example, it may be possible to combine parts such that CCA board space could be reduced and ultimately the need for an entire CCA might be eliminated. A reasonable reduction goal in this Parameter could be from 2 CCAs down to 1 CCA. The goal of 1 CCA, in this example, is listed under the column heading Parameter Try. In fact, this column contains a reasonable reduction in parameter value for five of the highest sensitivity parameters. The RF Module was eliminated from this exercise because it was unreasonable to reduce the parameter value below its baseline value.

Once the top five most Impactful Parameters were determined and reasonably achievable Parameter Try values were assigned the cost model could be run with ALL the potential improvements applied simultaneously and the corresponding impact to cost could be observed.

It is important to mention that no longer are the KSMs involved in the calculation. KSMs were only used to create a variation factor to understand the sensitivity, it was a theoretical adjustment. In this exercise, real values are being explored.

The two scenarios in Table 8 compare the Baseline Cost with the Try Cost. The Try Cost includes all the Parameter values from the Parameter Try column of Table 7 applied simultaneously.

	Value
Baseline System Cost	$8,464K
New System Cost (Try Cost)	$6,682K
Savings	$1,781K
% Improvement	21%

Table 8. Baseline vs. Potential Try Cost

By modifying the top five cost driving parameters from a baseline value to an achievable and improved value it is demonstrated that there would be a significant improvement to overall system cost. The results of the two scenarios are summarized in Table 8.

As can be seen in Table 8, the percentage improvement is 21% over the baseline which is a significant impact! Another way to interpret this result is in terms of return on investment (ROI). To modify a parameter value, it would of course be necessary to expend some resources to achieve the new value. For example, to reduce a design from 3 CCAs to 2 CCAs some amount of resources, or investment, must be made. To perform some amount of research, design, analysis, or trade study, there must be some expended resource which yields a parameter improvement. A system designer should know the cost of that expended resource. In this case if the system designer remains below a $1.7M investment then the project overall would demonstrate an improvement. In other words, a system designer could spend up to $1.7M to

Level 1	Level 2	Level 3	Block Name
001			Radar
	001.01		Antenna
		001.01.01	Radiator
		001.01.02	TR Product
		001.01.03	Duplexer
	001.02		Transmitter
		001.02.01	Power Amplifier
		001.02.02	Up Converter
		001.02.03	Local Oscillator
	001.03		Synchronizer
		001.03.01	Synchronizer
	001.04		Receiver
		001.04.01	Low Noise Amplifier
		001.04.02	Down Converter
		001.04.03	Local Oscillator
		001.04.04	IF Amplifier
		001.04.05	Filters
		001.04.06	2nd Down Converter
		001.04.07	2nd Local Oscillator
		001.04.08	Detector
		001.04.09	Analog to Digital Converter
	001.05		Processor
		001.05.01	Processor
	001.06		Power
		001.06.01	Transformer
		001.06.02	Rectifier
		001.06.03	Filter
		001.06.04	Regulator
	001.07		Display
		001.07.01	Video Amplifier
		001.07.02	Display

Table 9. Indentured System Numbering Structure

Level 1	Level 2	Level 3	Block Name
001			Radar
	001.01		Antenna
	001.01RU		Antenna Roll Up
		001.01.01	Radiator
		001.01.02	TR Product
		001.01.03	Duplexer
	001.02		Transmitter
	001.02RU		Transmitter Roll Up
		001.02.01	Power Amplifier
		001.02.02	Up Converter
		001.02.03	Local Oscillator
	001.03		Synchronizer
	001.03RU		Synchronizer Roll Up
		001.03.01	Synchronizer
	001.04		Receiver
	001.04RU		Receiver Roll Up
		001.04.01	Low Noise Amplifier
		001.04.02	Down Converter
		001.04.03	Local Oscillator
		001.04.04	IF Amplifier
		001.04.05	Filters
		001.04.06	2nd Down Converter
		001.04.07	2nd Local Oscillator
		001.04.08	Detector
		001.04.09	Analog to Digital Converter
	001.05		Processor
	001.05RU		Processor Roll Up
		001.05.01	Processor
	001.06		Power
	001.06RU		Power Roll Up
		001.06.01	Transformer
		001.06.02	Rectifier
		001.06.03	Filter
		001.06.04	Regulator
	001.07		Display
	001.07RU		Display Roll Up
		001.07.01	Video Amplifier
		001.07.02	Display

Table 10. Indentured Cost Numbering Structure

achieve improvements in those top five parameters which most significantly impact overall system cost. And of course, anything less than $1.7M contributes to profit margin. If those achievements could be realized there would be a 21% improvement in overall system cost which is clearly a significant improvement.

VI. Example: Sensitivity Algorithm Applied to "Real" Cost Model

With the cost sensitivity algorithm fully developed and understood, the effort now turns towards implementation of a "real" example. For the development of the cost sensitivity algorithm a sample cost model has been used. While this has led to a theoretical benefit, what remains to be seen is if this algorithm can be utilized in a more real-life example. To address and satisfy this question, a real example is available. In particular, a cost model based on the standardized WBS structure for an airborne RADAR system for a military aerospace application has been developed and can be utilized to test the cost sensitivity algorithm.

A. Selection of a "Real" Cost Model

In A System Engineering Approach Using Sensitivity Analysis For Reducing System Cost, JACP April 2022, an effort was made to consolidate block diagrams from a wide sample of available examples. This was done to create a generalized block diagram of an airborne RADAR for military applications and where each of the examples could be considered a subset of the more generalized form. The resulting block diagrams and definitions were then organized into a WBS structure (Table 9).

The generalized WBS structure was shown useful as a foundation for both a system model and a cost model. As can be seen, the suggested cost model (Table 10) is the same as the WBS in that it maintains the same structure however the cost model includes additional rows for "Roll Up." A cost tool could call out Level 2 hardware, for example an antenna. However, an antenna is also a collection of Level 3 hardware blocks. In this case, both options are included in the cost model. And when the model is run to produce an estimate either, but not both would be selected.

This new RADAR cost model represents a real-life example of a RADAR cost model upon which to verify the benefits of employing the cost sensitivity algorithm.

B. Impactful Parameters & KSM Values

The work of determining the Impactful Parameters has already been completed. The same set of Impactful Parameters which were previously used and appear in Table 4 will be used once again in this analysis. No additional work in this area is required.

The work of determining the KSM values has already been completed. The same set of KSM

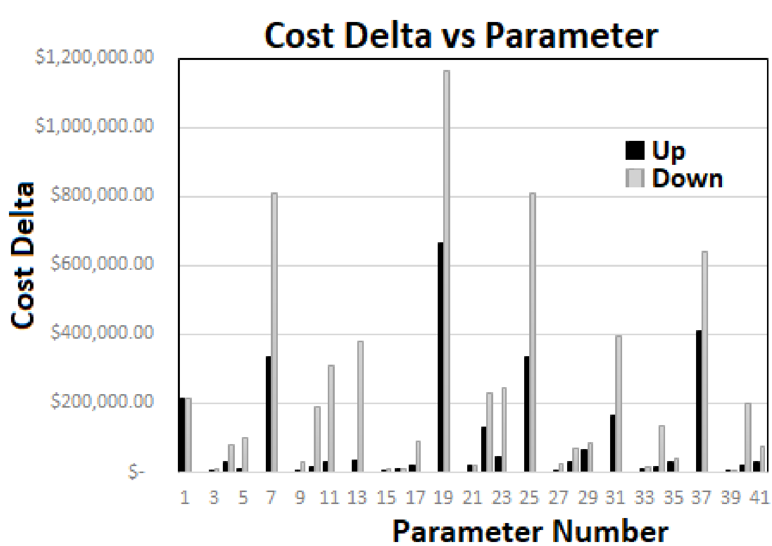


Figure 3 - Real-Life Cost Delta vs. Parameter

Parameter	Component	Sensitivity or Delta "mid"	Rank Of Impact
Clock Speed (USD/ MHz)	Antenna	$53.59K	17
	Display	$52.39K	18
	Power	$36K	20
	Processor	$76.72K	13
	Receiver	$144.95K	10
	Synchronizer	$54.75K	15
	Transmitter	$169.79K	9
Discreet Comp per PCB (USD/n)	Antenna	$6K	26
	Display	$2.08K	28
	Power	$13.58K	24
	Processor	$14.63K	23
	Receiver	$21.52K	21
	Synchronizer	$4.29K	27
	Transmitter	$16.08K	22
Integrated Comp per PCB (USD/n)	Antenna	$5.46K	16
	Display	$109.04K	11
	Power	$73.66K	14
	Processor	$48.07K	19
	Receiver	$180.15K	8
	Synchronizer	$12.15K	25
	Transmitter	$102.36K	12
PCB Size (USD/in2)	Antenna	0	29
	Display	0	41
	Power	0	39
	Processor	0	37
	Receiver	0	35
	Synchronizer	0	33
	Transmitter	0	31
Total CCAs (USD/n)	Antenna	$216.25K	6
	Display	$525.89K	4
	Power	$279.63K	5
	Processor	$573.07K	2
	Receiver	$913.70K	1
	Synchronizer	$207.09K	7
	Transmitter	$571.00K	3

Table 11. Real-Life Cost Model Results Sorted by Parameter

values which were previously used and appear in Table 5 will be used once again in this analysis. No additional work in this area is required.

C. Re-Run of Algorithm & Data Collection

As before, an Excel file was created for the real-life cost model analysis which shows all the components with their respective parameters and their variation factors (Table 17). The table was again used to collect and organize the information including hardware components, Impactful Parameters, KSM values, etc. As before, the spread sheet calculates the vary "up" and vary "down" values for the parameters. To establish a baseline cost, the cost model was run to estimate the cost for the overall system and the results were compiled.

The cost tool was then run repeatedly, and for each consecutive run, only one parameter from the list was changed keeping all other values in their baseline condition. The overall system cost was then collected for each permutation of parameter value. Using the results of each run from the cost tool, the delta "mid" and delta "range" values were calculated for each parameter. With the delta "mid" values calculated, the values could be ranked in order of greatest to least impact to overall system cost. The ranking appears in the table and is color coded, as before, with the ten most Impactful Parameters as red, the next ten as yellow, the next ten as green, and the remainder as uncolored (Table 17).

D. Results - Algorithm on "Real" Cost Model

The results are graphically illustrated in Figure 3. As before, the various parameters have a wide variety of impact to overall system cost.

The results were sorted by parameter (Table 11). It is very clear from the table that for any parameter, there exists a set of components upon which the parameter applies. And, for each of the components, there is clearly a difference in the sensitivity of that parameter depending upon to which component it is applied. In the case of

Parameter	Variation Factor	Component	Sensitivity or Delta "mid"
Total CCA	20%	Receiver	$913.71K USD/n
Integrated Components per PCB	5%	Receiver	$180.15K USD/n
Clock Speed	20%	Transmitter	$169.79K USD/MHz
Discreet Components per PCB	10%	Receiver	$21.52K USD/n
PCB Size	40%	Antenna	$0K USD/

Table 12. Real-Life Parameter Maximums

Clock Speed, for example, the parameter is associated with seven different components. The Receiver component has the highest sensitivity for this parameter, and in addition, ranks as 10th most Impactful Parameter in sensitivity for the entire system.

In Table 12, the maximums for each parameter are collected. In other words, for the parameter of Total CCAs, it was determined in Table 11 that of the seven components, Receiver had the highest sensitivity and is in fact ranked as 1st overall. Therefore, in Table 12, for the parameter of Total CCAs, only the component Receiver is listed with its corresponding sensitivity, or delta "mid" value. The same logic applies for all the other parameters listed in Table 12.

In addition, all cost scenarios were numbered in order of overall impact to system cost (Table 17, column R). Included are the corresponding components along with the parameter which influences the component costs. As mentioned, the ranking was color coded with red for the highest impact or sensitivity (1-10), yellow for medium (11-20) and green for low (21-30). It can be seen, for example, that the total number of CCAs in the Receiver has the highest cost sensitivity and was ranked correspondingly with a value of 1.

E. Multivariable Analysis & Trade Studies

A robust structured system modelling approach utilizes the concept of modularity. If the system is comprised of modules, then the possibility exists where modules could be swapped to modify the system for various performance characteristics. At the same time, if the cost model mirrors the system model, then as the system is being defined, a rough cost estimation could be determined simultaneously.

Even with a modular approach, when designing a system more than one variable must be considered. Choices are made regarding those variables. In most cases, variable choices have competing impacts. For example, one design architecture may have "better" performance using more power vs. "worse" performance using less power. Decisions for a sub-system need to be evaluated at a system level. A system designer needs to consider the design as a system and realize that any change potentially has an impact beyond the sub-system. It is not usually possible to make an architecture or hardware change irrespective of the larger view of the system. This is really the heart of system engineering, consideration of an entire system, not just a collection of sub-system parts.

This is particularly important when considering cost because it is not possible to swap out cost as modular blocks and estimate new costs without understanding that there are affects to the system. There are multilevel impacts when modular blocks are substituted. Simply swapping out a block and estimating cost gives a first order indication of the cost impact. But until the design is finalized it is only a rough estimate. There is a spiral approach to design. As choices are made, impacts are assessed, costs can be estimated, new choices are made, and eventually the design spirals into a solution.

To decide between competing variables a trade study can be employed. A trade study is a useful tool which allows a designer to compare and contrast the various possible choices to determine which solution would be "best" for the given application.

This work is intended as a tool for a cost analyst to provide a designer a "compass" on where to focus attention by identifying the components with the highest cost sensitivities.

VII. Return on Investment (ROI): "Real" Cost Model

Consistent with the sample cost model example, all the calculations and results presented remain of theoretical importance. However, the value of this work lies in the application of the results. The question which is of primary concern relates to how a system architecture can be optimized in terms of performance and cost early in the life cycle of a program. To illustrate the significance of the results upon potential improvements to the real-life system, the following discussion is offered.

The information which appears in Table 17 includes the top 29 most Impactful Parameters. What was discovered was that a few parameters have the greatest sensitivity affecting the cost of the overall system. It is unrealistic to consider improvements to a system in such a broad number of parameters over a broad set of components. Instead, focus will be applied to a more conservative subset. What remains to be done is to modify a reasonable set of those Impactful Parameters to see realistically how it will affect system cost. In Table 13, three options are presented: option A represents a reasonable and achievable change in hardware, option B represents improvements to the top three parameters, and option C represents improvements to the top five cost driving parameters.

The column Parameter "Was" indicates the value of the Parameter which was used in the baseline cost calculations. The value in the "Try" column

Option A: Was/Try, Reasonable Expectation

Component Level 2	Parameter	Rank of impact	Parameter "Was"	Parameter "Try"
Receiver	Total CCAs	**1**	3	2
Processor	Total CCAs	**2**	2	1
Receiver	Integrated Components per PCB	**8**	45	42
Transmitter	Clock Speed	**9**	400	300
Receiver	Clock Speed	**10**	250	300

Option B: Was/Try, Multiple Teams

Component Level 2	Parameter	Rank of impact	Parameter "Was"	Parameter "Try"
Receiver	Total CCAs	**1**	3	2
Receiver	Integrated Components per PCB	**8**	45	42
Transmitter	Clock Speed	**9**	400	300

Option C: Was/Try, Top 5

Component Level 2	Parameter	Rank of impact	Parameter "Was"	Parameter "Try"
Receiver	Total CCAs	**1**	3	2
Processor	Total CCAs	**2**	2	1
Transmitter	Total CCAs	**3**	2	1
Display	Total CCAs	**4**	4	3
Power	Total CCAs	**5**	2	1

Table 13. Was/Try Parameter Value Options

contains a reasonable modification to parameter value. In other words, with some reasonable effort, it may be an achievable goal to modify the "was" to the "try" value.

Option A is a reasonable effort. This is an option which if undertaken, it may result in achieving these goals. Options B & C are not very realistic. Option B, for example, because it involved three distinct parameters, it would require three separate disciplinary teams. While option C indiscriminately selects the top five drivers and is hardly likely to be achievable. Still, options B & C help to bound the possible improvements.

The cost model could be run with ALL of the potential improvements applied simultaneously and the corresponding impact to cost could be observed. It is important to mention that no longer are the KSMs involved in the calculation. KSMs were only used to create a variation factor to understand the sensitivity, it was a theoretical adjustment. In this exercise, real values are being explored.

By modifying the cost driving parameters from a baseline value to an achievable and improved value (Table 17), it is demonstrated that there would be a significant improvement to overall system cost. Table 14 is a summary of the system cost result when the options are exercised.

As can be seen in Table 14, the percentage improvements are 23%, 13% and 41% over the baseline which is a significant impact! Of course, as mentioned, option A is really the only option under consideration and has a 23% potential improvement in cost.

Another way to interpret this result is in terms of return on investment (ROI). To modify a parameter value, it would of course be necessary to expend some resources to achieve the new value. For example, to reduce a design from 3 CCAs to 2 CCAs some amount of resources, or investment, must be made. There must be some amount of research, design, analysis, or trade study. There must be some expended resource which yields a parameter improvement. A system designer should know the cost of that expended resource. In this case, if the system designer remains below a $3.9M investment then the project, overall, would demonstrate an improvement. In other words, a system designer could spend up to $3.9M to achieve improvements in those parameters for option A which most significantly impact overall system cost. And of course, anything less than $3.9M contributes to profit margin. If those achievements could be realized there would be a 23% improvement in overall system cost which is clearly a significant improvement.

Option A: Was/Try, Reasonable Expectation

$16,975,050	<-- Baseline System Cost
$13,080,099	<-- New Improved System Cost
$3,894,951	<-- Savings
23%	<-- % Improvement

Option B: Was/Try, Multiple Teams

$16,975,050	<-- Baseline System Cost
$14,799,474	<-- New Improved System Cost
$2,175,576	<-- Savings
13%	<-- % Improvement

Option C: Was/Try, Top 5

$16,975,050	<-- Baseline System Cost
$10,008,258	<-- New Improved System Cost
$6,966,793	<-- Savings
41%	<-- % Improvement

Table 14. Was/Try Parameter Value Cost Results

VIII. Conclusion

This paper documents the application of a cost sensitivity algorithm upon the various components in a system to analyze and determine which subsystem components in a chosen design solution have the highest sensitivity to overall cost. This paper highlights the areas to which a system designer could apply focus to reduce the overall system cost early in the life cycle of a program. It was shown using sensitivity analysis that a cost sensitivity algorithm was developed including a discussion on key size metrics. It was shown the cost sensitivity algorithm was applied to a sample cost model and that it demonstrates which component parameters were most sensitive and the biggest cost drivers in the system design. In addition, an alternative was suggested which offered the system designer a significant opportunity to improve cost. A return on investment (ROI) was calculated using the result to suggest a trade study budget for achieving the potential cost improvements. The fully developed cost sensitivity algorithm with KSMs was then applied to a "real" example cost model based on a generalized work breakdown structure (WBS) for a RADAR system applied to military applications in the aerospace industry. The result of applying the algorithm was presented and demonstrates the significant impact achievable to overall system cost when focus is applied appropriately to the areas for which overall system cost is most sensitive. And finally, for the "real" example, the algorithm highlights the areas where trade studies could be performed and yields a target return on investment (ROI) budget to achieve those cost improvements at the beginning of the life cycle of a program.

References

1. Corporate Finance Institute, "Normalization: Adjusting Financial Statements," https://corporatefinanceinstitute.com/resources/knowledge/finance/financial-statement-normalization/, 2020.

2. United States Department of Agriculture, "Normalized Prices," https://www.ers.usda.gov/data-products/normalized-prices/, 2019.

3. Mislick, Gregory K; Nussbaum, Daniel A;, "Cost Estimation: Methods and Tools. Ch5 Data Normalization," in Cost Estimation: Methods and Tools. Ch5 Data Normalization, Wiley, 2015.

4. ICEAA Cost Estimating Body of Knowledge Module 4, "Data Collection and Normalization," https://www.iceaaonline.com/cebok.

5. Corporate Finance Institute, "Cost Driver: The direct cause of a cost," https://corporatefinanceinstitute.com/resources/knowledge/accounting/cost-driver/.

6. M. E. Porter, Competitive Advantage: Creating and Sustaining Superior Performance, Free Press, 1998.

7. Federal Aviation Administration, NAS System Engineering Manual, Ver 3.1, Air Traffic Organization, 2006.

8. TAMU, "Trade Studies: Finding the best alternative for system performance from available options," https://engineering.tamu.edu/media/2225601/Trade-Studies.pdf, College Station.

9. Felix, Art, "Standard Approach to Trade Studies: A Process Improvement Model that Enables Systems Engineers to Provide Information to the Project Manager by Going Beyond the Summary Matrix," in INCOSE Mid-Atlantic Regional Conference, 2004.

10. Felix, A, "Standard Approach to Trade Studies," in INCOSE 2004 - 14th Annual International Symposium Proceedings, 2004.

11. Cilli, Matthew V.; Parnell, Gregory S.;, "Systems Engineering Tradeoff Study Process Framework," INCOSE, p. 17, 2014.

12. V. Belton and T. J. Stewart, "Multiple Criteria Decision Analysis," Springer-Science+Business Media, B.V., 2002.

13. C. Zopounidis and P. M. Pardalos, Handbook Of Multicriteria Analysis, Springer, 2010.

14. T. L. Saaty, "How to make a decision: The Analytic Hierarchy Process," European Journal of Operational Research, vol. 48, pp. 9-26, 1990.

15. T. L. Saaty, "The Analytic Network Process".

16. Forrester, Jay W.; Breierova, Lucia; Choudhari, Mark, "An Introduction to Sensitivity Analysis," MIT System Dynamics in Education Project, p. 64, 6 September 1996.

17. Bingulac, S, "Minimum Sensitivity Adaptive Systems," Science Direct, 1966.

18. D. J. Pannell, "Sensitivity Analysis of Normative Economic Models: Theoretical Framework and Practical Strategies," Agricultural Economics, 1997.

19. P. Glasserman and S. Tayur, "Sensitivity Analysis for Base-Stock Levels in Multiechelon Production-Inventory Systems," Management Science, vol. 41, no. 2, pp. 263-281, 1995.

20. B. Iooss and A. Saltelli, "Introduction: Sensitivity Analysis," Chatou, France.

21. L. Nurhadi, S. Boren and H. Ny, "A sensitivity analysis of total cost of ownership for electric public bus transport systems in Swedish medium sized cities," Transportation Research Procedia, vol. 3, pp. 818-827, 2014.

22. A. MacGillivray, H. Jeffrey, M. Winskel and I. Bryden, "Innovation and cost reduction for marine renewable energy: A learning investment sensitivity analysis," Technological Forecasting & Social Change: An International Journal, vol. 87, pp. 108-124, 2014.

23. PRICE Systems, "Processes & Best Practices: Design-to-Cost Target," https://www.pricesystems.com/cost-estimating-process-integrations/, 2020.

24. COSYSMO, "Constructive Systems Engineering Cost Model," http://cosysmo.mit.edu/, 2020.

25. Galorath, "SEER for Hardware, Electronics and Systems," https://galorath.com/our-products/, 2020.

26. Polidi, Danny; Crist, Michael, "Foundation Of Structured Architecture, System & Cost Modeling," in ICEAA Symposium 2020, Dallas, 2018.

27. Corporate Finance Institute, "What is Sensitivity Analysis," https://corporatefinanceinstitute.com/resources/knowledge/modeling/what-is-sensitivity-analysis/.

28. V. Fragoulakis and C. Mitropoulou, Economic Evaluation in Genomic Medicine, Academic Press, 2015.

29. Polidi, Danny; Bloom, David, "Building A Complex Hardware Cost Model For Antennas," ICEAA 2014 Professional Development & Training Workshop, Denver, 2014.

Appendix

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		Hardware		Parameters					Calculated System Cost					Interpretation of Cost Data					
2	Count	Sub System Block	Component	Parameter	Prmtr Value	Vary Amt	Varied "Up" Prmtr Value	Varied "Down" Prmtr Value	blank	Baseline Cost	Varied "up" cost	Varied "Down" cost	blank	delta "up" cost	delta "down" cost	delta "mid"	delta "range"	Rank of impact	Color Rank
3		Receiver Module																	
4			Receiver																
5	1		Receiver	Total CCAs	2	20%	2.4	1.6		8464002	8635812	8292434		171810	171568	171689	121	3	RED
6	2		Receiver	PCB Size	30	20%	36	24		8464002	8464234	8463702		231	300	266	34	30	GRN
7	3		Receiver	Discreet Components per PCB	40	20%	48	32		8464002	8465383	8460653		1381	3350	2365	984	23	GRN
8	4		Receiver	Integrated Components per PCB	45	20%	54	36		8464002	8554945	8355278		90943	108724	99834	8891	6	RED
9	5		Receiver	Clock Speed	240	20%	288	192		8464002	8470485	8454122		6483	9880	8181	1699	16	YEL
10			RF Module							8464002								39	
11	6		RF Module	Total CCAs	0.5	20%	0.6	0.4		8464002	8532808	8368659		68806	95343	82074	13269	7	RED
12	7		RF Module	PCB Size	30	20%	36	24		8464002	8465564	8462440		1562	1562	1562	0	25	GRN
13	8		RF Module	Discreet Components per PCB	6	20%	7	5		8464002	8464373	8463262		370	740	555	185	27	GRN
14	9		RF Module	Integrated Components per PCB	4	20%	5	3		8464002	8480703	8446047		16700	17955	17328	627	13	YEL
15	10		RF Module	Clock Speed	800	20%	960	640		8464002	8471924	8457594		7922	6408	7165	757	17	YEL
16			RF Machined Housing							8464002								40	
17	11		RF Machined Housing	Weight	3.5	20%	4.2	2.8		8464002	8482248	8445424		18245	18578	18412	166	12	YEL
18	12		RF Machined Housing	Volume	0.4	20%	0.48	0.32		8464002	8464044	8463938		42	64	53	11	34	
19			Rcv Chassis							8464002								41	
20	13		Rcv Chassis	Weight	15	20%	18	12		8464002	8477620	8439277		13618	24725	19171	5554	11	YEL
21	14		Rcv Chassis	Volume	4	20%	4.8	3.2		8464002	8464007	8463998		4	4	4	0	37	
22		Digitial Processing								8464002								42	
23			Converter & Noise Reduction							8464002								43	
24	15		Converter & Noise Reduction	Total CCAs	3	20%	3.6	2.4		8464002	8906581	8338728		442579	125275	283927	158652	2	RED
25	16		Converter & Noise Reduction	PCB Size	30	20%	36	24		8464002	8464444	8463561		442	442	442	0	28	GRN
26	17		Converter & Noise Reduction	Discreet Components per PCB	35	20%	42	28		8464002	8470804	8462064		6802	1939	4370	2432	20	YEL
27	18		Converter & Noise Reduction	Integrated Components per PCB	120	20%	144	96		8464002	8598094	8303686		134091	160316	147204	13112	4	RED
28	19		Converter & Noise Reduction	Clock Speed	20	20%	24	16		8464002	8465109	8442527		1107	21475	11291	10184	14	YEL
29			Data Processing							8464002								44	
30	20		Data Processing	Total CCAs	2	20%	2.4	1.6		8464002	8669292	7967018		205289	496985	351137	145848	1	RED
31	21		Data Processing	PCB Size	30	20%	36	24		8464002	8464402	8463640		400	362	381	19	29	GRN
32	22		Data Processing	Discreet Components per PCB	250	20%	300	200		8464002	8466173	8460964		2171	3038	2604	434	22	GRN
33	23		Data Processing	Integrated Components per PCB	750	20%	900	600		8464002	8601269	8377674		137267	86328	111798	25469	5	RED
34	24		Data Processing	Clock Speed	20	20%	24	16		8464002	8474208	8451898		10206	12104	11155	949	15	YEL
35			Purchased Memory							8464002								45	
36	25		Purchased Memory	Total CCAs	1	20%	1.2	0.8		8464002	8465693	8462322		1691	1681	1686	5	24	GRN
37	26		Purchased Memory	PCB Size	30	20%	36	24		8464002	8464015	8463989		13	13	13	0	35	
38	27		Purchased Memory	Discreet Components per PCB	55	20%	66	44		8464002	8464195	8463720		193	282	237	44	31	
39	28		Purchased Memory	Integrated Components per PCB	43	20%	52	34		8464002	8465579	8462505		1577	1497	1537	40	26	GRN
40	29		Purchased Memory	Clock Speed	20	20%	24	16		8464002	8464015	8463783		13	219	116	103	32	
41			Interconnect - Data Bus							8464002								46	
42	30		Interconnect - Data Bus	Total CCAs	1	20%	1.2	0.8		8464002	8515076	8413228		51073	50774	50924	150	8	RED
43	31		Interconnect - Data Bus	PCB Size	30	20%	36	24		8464002	8464063	8463941		61	61	61	0	33	
44	32		Interconnect - Data Bus	Discreet Components per PCB	55	20%	66	44		8464002	8468355	8456537		4353	7465	5909	1556	19	YEL
45	33		Interconnect - Data Bus	Integrated Components per PCB	43	20%	52	34		8464002	8494482	8430804		30480	33198	31839	1359	9	RED
46	34		Interconnect - Data Bus	Clock Speed	20	20%	24	16		8464002	8467464	8460131		3462	3871	3667	204	21	GRN
47			Instrumentation Panel							8464002								47	
48	35		Instrumentation Panel	Weight	2.5	20%	3	2		8464002	8470998	8456819		6996	7183	7090	94	18	YEL
49	36		Instrumentation Panel	Volume	0.4	20%	0.48	0.32		8464002	8464002	8464002		0	0	0	0	38	
50			Digital Processing Chassis							8464002								48	
51	37		Digital Processing Chassis	Weight	18	20%	21.6	14.4		8464002	8481946	8437212		17944	26790	22367	4423	10	RED
52	38		Digital Processing Chassis	Volume	4	20%	4.8	3.2		8464002	8464007	8463997		5	5	5	0	36	

Table 15. Full Data Set Using 20% Variation Factor

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		**Hardware**		**Parameters**						**Calculated System Cost**				**Interpretation of Cost Data**					
2	Count	Sub System Block	Component	Parameter	Prmtr Value	Vary Amt (KSM)	Varied "Up" Prmtr Value	Varied "Down" Prmtr Value	blank	Baseline Cost	Varied "up" cost	Varied "Down" cost	blank	delta "up" cost	delta "down" cost	delta "mid"	delta "range"	Rank of impact	Color Rank
3		Receiver Module																	
4			Receiver																
5	1		Receiver	Total CCAs	2	20%	2.4	1.6		8464002	8635812	8292434		171810	171568	171689	121	3	RED
6	2		Receiver	PCB Size	30	65%	49.5	10.5		8464002	8464754	8463050		752	952	852	100	26	GRN
7	3		Receiver	Discreet Components per PCB	40	5%	42	38		8464002	8463608	8462425		-394	1577	591	986	28	GRN
8	4		Receiver	Integrated Components per PCB	45	5%	48	42		8464002	8513752	8436029		49750	27973	38862	10888	9	RED
9	5		Receiver	Clock Speed	240	20%	288	192		8464002	8470485	8454122		6483	9880	8181	1699	17	YEL
10			RF Module							8464002								39	
11	6		RF Module	Total CCAs	0.5	20%	0.6	0.4		8464002	8532808	8368659		68806	95343	82074	13269	4	RED
12	7		RF Module	PCB Size	30	65%	49.5	10.5		8464002	8469078	8458926		5076	5077	5076	0	19	YEL
13	8		RF Module	Discreet Components per PCB	6	5%	7	5		8464002	8464373	8463262		370	740	555	185	29	GRN
14	9		RF Module	Integrated Components per PCB	4	5%	5	3		8464002	8480703	8446047		16700	17955	17328	627	13	YEL
15	10		RF Module	Clock Speed	800	20%	960	640		8464002	8471924	8457594		7922	6408	7165	757	18	YEL
16			RF Machined Housing							8464002								40	
17	11		RF Machined Housing	Weight	3.5	60%	5.6	1.4		8464002	8517543	8423335		53541	40667	47104	6437	8	RED
18	12		RF Machined Housing	Volume	0.4	75%	0.7	0.1		8464002	8464120	8463741		118	261	189	72	31	
19			Rcv Chassis							8464002								41	
20	13		Rcv Chassis	Weight	15	60%	24	6		8464002	8514311	8397695		50309	66308	58308	7999	6	RED
21	14		Rcv Chassis	Volume	4	75%	7	1		8464002	8464018	8463986		16	16	16	0	37	
22		Digitial Processing								8464002								42	
23			Converter & Noise Reduction							8464002								43	
24	15		Converter & Noise Reduction	Total CCAs	3	20%	3.6	2.4		8464002	8906581	8338728		442579	125275	283927	158652	2	RED
25	16		Converter & Noise Reduction	PCB Size	30	65%	49.5	10.5		8464002	8465683	8462709		1681	1293	1487	194	24	GRN
26	17		Converter & Noise Reduction	Discreet Components per PCB	35	5%	37	33		8464002	8464556	8463448		554	554	554	0	30	GRN
27	18		Converter & Noise Reduction	Integrated Components per PCB	120	5%	126	114		8464002	8542729	8470211		78727	-6209	36259	42468	10	RED
28	19		Converter & Noise Reduction	Clock Speed	20	20%	24	16		8464002	8465109	8442527		1107	21475	11291	10184	15	YEL
29			Data Processing							8464002								44	
30	20		Data Processing	Total CCAs	2	20%	2.4	1.6		8464002	8669292	7967018		205289	496985	351137	145848	1	RED
31	21		Data Processing	PCB Size	30	65%	49.5	10.5		8464002	8465301	8462880		1298	1122	1210	88	25	GRN
32	22		Data Processing	Discreet Components per PCB	250	5%	263	237		8464002	8464245	8462891		243	1111	677	434	27	GRN
33	23		Data Processing	Integrated Components per PCB	750	5%	788	712		8464002	8517175	8460534		53173	3468	28320	24853	12	YEL
34	24		Data Processing	Clock Speed	20	20%	24	16		8464002	8474208	8451898		10206	12104	11155	949	16	YEL
35			Purchased Memory							8464002								45	
36	25		Purchased Memory	Total CCAs	1	20%	1.2	0.8		8464002	8465693	8462322		1691	1681	1686	5	22	GRN
37	26		Purchased Memory	PCB Size	30	65%	49.5	10.5		8464002	8464045	8463970		43	32	37	6	35	
38	27		Purchased Memory	Discreet Components per PCB	55	5%	58	52		8464002	8464059	8463904		57	98	77	20	34	
39	28		Purchased Memory	Integrated Components per PCB	43	5%	52	34		8464002	8465579	8462505		1577	1497	1537	40	23	GRN
40	29		Purchased Memory	Clock Speed	20	20%	24	16		8464002	8464015	8463783		13	219	116	103	33	
41			Interconnect - Data Bus							8464002								46	
42	30		Interconnect - Data Bus	Total CCAs	1	20%	1.2	0.8		8464002	8515076	8413228		51073	50774	50924	150	7	RED
43	31		Interconnect - Data Bus	PCB Size	30	65%	49.5	10.5		8464002	8464200	8463855		198	147	173	25	32	
44	32		Interconnect - Data Bus	Discreet Components per PCB	55	5%	58	52		8464002	8465471	8461479		1469	2523	1996	527	21	GRN
45	33		Interconnect - Data Bus	Integrated Components per PCB	43	5%	52	34		8464002	8494482	8430804		30480	33198	31839	1359	11	YEL
46	34		Interconnect - Data Bus	Clock Speed	20	20%	24	16		8464002	8467464	8460131		3462	3871	3667	204	20	YEL
47			Instrumentation Panel							8464002								47	
48	35		Instrumentation Panel	Weight	2.5	60%	4	1		8464002	8484563	8456765		20561	7237	13899	6662	14	YEL
49	36		Instrumentation Panel	Volume	0.4	75%	0.7	0.1		8464002	8464003	8464002		0	0	0	0	38	
50			Digital Processing Chassis							8464002								48	
51	37		Digital Processing Chassis	Weight	18	60%	28.8	7.2		8464002	8524752	8388856		60750	75146	67948	7198	5	RED
52	38		Digital Processing Chassis	Volume	4	75%	7	1		8464002	8464021	8463984		18	18	18	0	36	

Table 16. Full Data Set Using KSM Variation Factors

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1		Hardware		Parameters							Calculated System Cost				Interpretation of Cost Data					
2	Count	Level 1	WBS	Level 2	Parameter	Prmtr Value	Vary Amt (KSM)	Varied "Up" Prmtr Value	Varied "Down" Prmtr Value	blank	Baseline Cost	Varied "up" cost	Varied "Down" cost	blank	delta "up" cost	delta "down" cost	delta "mid"	delta "range"	Rank of impact	Color Rank
3		RADAR	001.01	Antenna																
4	1	RADAR		Antenna	Total CCAs	2	20%	2.4	1.6		16975050	17192074	16759573		217024	215478	216251	773	6	RED
5	2	RADAR		Antenna	PCB Size	30	65%	49.5	10.5		16975050	16975050	16975050		0	0	0	0	29	GRN
6	3	RADAR		Antenna	Discreet Components per PCB	48	5%	51	45		16975050	16976386	16964376		1336	10674	6005	4669	26	GRN
7	4	RADAR		Antenna	Integrated Components per PCB	40	5%	42	38		16975050	17052589	16943311		77539	31739	54639	22900	16	YEL
8	5	RADAR		Antenna	Clock Speed	400	20%	480	320		16975050	16984776	16877596		9726	97454	53590	43864	17	YEL
9		RADAR	001.02	Transmitter							16975050								30	GRN
10	6	RADAR		Transmitter	Total CCAs	2	20%	2.4	1.6		16975050	17308905	16166897		333855	808153	571004	237149	3	RED
11	7	RADAR		Transmitter	PCB Size	30	65%	49.5	10.5		16975050	16975050	16975050		0	0	0	0	31	
12	8	RADAR		Transmitter	Discreet Components per PCB	55	5%	58	52		16975050	17004571	16972409		29521	2642	16081	13440	22	GRN
13	9	RADAR		Transmitter	Integrated Components per PCB	35	5%	37	33		16975050	17162777	16958067		187727	16983	102355	85372	12	YEL
14	10	RADAR		Transmitter	Clock Speed	400	20%	480	320		16975050	17003903	16664333		28852	310717	169785	140932	9	RED
15		RADAR	001.03	Synchronizer							16975050								30	GRN
16	11	RADAR		Synchronizer	Total CCAs	1	20%	1.2	0.8		16975050	17009736	16595557		34685	379493	207089	172404	7	RED
17	12	RADAR		Synchronizer	PCB Size	30	65%	49.5	10.5		16975050	16975050	16975050		0	0	0	0	33	
18	13	RADAR		Synchronizer	Discreet Components per PCB	40	5%	42	38		16975050	16975766	16967186		715	7864	4290	3574	27	GRN
19	14	RADAR		Synchronizer	Integrated Components per PCB	10	5%	11	9		16975050	16987219	16962910		12168	12140	12154	14	25	GRN
20	15	RADAR		Synchronizer	Clock Speed	2000	20%	2400	1600		16975050	16995861	16886362		20810	88688	54749	33939	15	YEL
21		RADAR	001.04	Receiver							16975050								30	GRN
22	16	RADAR		Receiver	Total CCAs	3	20%	3.6	2.4		16975050	17638334	15810925		663284	1164125	913705	250420	1	RED
23	17	RADAR		Receiver	PCB Size	20	65%	33	7		16975050	16975050	16975050		0	0	0	0	35	
24	18	RADAR		Receiver	Discreet Components per PCB	60	5%	63	57		16975050	16996058	16953025		21007	22025	21516	509	21	GRN
25	19	RADAR		Receiver	Integrated Components per PCB	45	5%	48	42		16975050	17205498	16845203		230448	129847	180147	50300	8	RED
26	20	RADAR		Receiver	Clock Speed	250	20%	300	200		16975050	17020973	16731077		45923	243973	144948	99025	10	RED
27		RADAR	001.05	Processor							16975050								30	GRN
28	21	RADAR		Processor	Total CCAs	2	20%	2.4	1.6		16975050	17310613	16164481		335562	810569	573066	237503	2	RED
29	22	RADAR		Processor	PCB Size	30	65%	49.5	10.5		16975050	16975050	16975050		0	0	0	0	37	
30	23	RADAR		Processor	Discreet Components per PCB	48	5%	51	45		16975050	16978503	16949246		3453	25804	14629	11176	23	GRN
31	24	RADAR		Processor	Integrated Components per PCB	40	5%	42	38		16975050	17043041	16946892		67990	28158	48074	19916	19	YEL
32	25	RADAR		Processor	Clock Speed	800	20%	960	640		16975050	17042420	16888982		67370	86068	76719	9349	13	YEL
33		RADAR	001.06	Power							16975050								30	GRN
34	26	RADAR		Power	Total CCAs	2	20%	2.4	1.6		16975050	17138839	16579579		163789	395471	279630	115841	5	RED
35	27	RADAR		Power	PCB Size	20	65%	33	7		16975050	16975050	16975050		0	0	0	0	39	
36	28	RADAR		Power	Discreet Components per PCB	80	5%	84	76		16975050	16991472	16964320		16422	10730	13576	2846	24	GRN
37	29	RADAR		Power	Integrated Components per PCB	30	5%	32	28		16975050	17109809	16962497		134759	12553	73656	61103	14	YEL
38	30	RADAR		Power	Clock Speed	50	20%	60	40		16975050	17005124	16933110		30074	41941	36007	5933	20	YEL
39		RADAR	001.07	Display							16975050								30	GRN
40	31	RADAR		Display	Total CCAs	4	20%	4.8	3.2		16975050	17615069	16563280		640018	411770	525894	114124	4	RED
41	32	RADAR		Display	PCB Size	20	65%	33	7		16975050	16975050	16975050		0	0	0	0	41	
42	33	RADAR		Display	Discreet Components per PCB	28	5%	30	26		16975050	16975570	16971410		520	3640	2080	1560	28	GRN
43	34	RADAR		Display	Integrated Components per PCB	35	5%	37	33		16975050	17175029	16956949		199979	18101	109040	90939	11	YEL
44	35	RADAR		Display	Clock Speed	100	20%	120	80		16975050	17006207	16901432		31157	73618	52387	21231	18	YEL

Table 16. Full Data Set - Algorithm Using "Real" System Cost Model

*Mr. **Danny Polidi** received a B.S. and M.S. degree in Electrical Engineering from the California Polytechnic State University, San Luis Obispo, CA, and a PhD for Systems Engineering at CSU. Upon graduation, started working at Space Systems/Loral on high frequency, microwave designs for space applications. Later, at Radian Technology, he became the Product Manager of the Digitally Tuned Oscillator Product Line where he was responsible for designing new circuits, writing code, and production. At NANOmetrics, Danny managed all Electronic Engineering activities. From 2004 – present he has worked at Raytheon as a Section Manger, Team Lead, Cost Account Manager and has been certified as a Program Manager.*

***Mike Crist** received his B.S. in Computer Science from Embry-Riddle Aeronautical University in 2003 and his M.S. in Electrical Engineering from The University of Texas at Dallas in 2005. He is currently enrolled at Colorado State University working on a Ph.D in Systems Engineer. Mike has had a variety of roles over his 20+ year career, including embedded software developer, FPGA developer, circuit card designer, cost account manager, integrated product team lead and engineering tool strategist. He currently works on staff for an Electrical / Mechanical Design Center on various model based engineering projects.*

***V. Chandrasekar** (S'83–M'87–F'03) received the bachelor's degree from IIT Kharagpur, Kharagpur, India, and the Ph.D. degree from Colorado State University (CSU), Fort Collins, CO, USA. He has been a Visiting Professor with the National Research Council of Italy, Rome, Italy, the University of Helsinki, Helsinki, Finland, the Finnish Meteorological Institute, Helsinki, IIT Kharagpur, and Indian Institute of Science,, an Affiliate Scientist with the National Aeronautics and Space Administration (NASA)'s Jet Propulsion Laboratory, Pasadena, CA, USA, a Distinguished Visiting Scientist with the NASA Goddard Space Flight Center, Greenbelt, MD, USA, and a Distinguished Professor of Finland (FiDiPro). He has also been a Director of the Research Experiences for Undergraduate Program for over 25 years, where he is involved in promoting research in the undergraduate curriculum. He is currently a University Distinguished Professor with CSU. He is also the Research Director of the National Science Foundation Engineering Research Center for Collaborative Adaptive Sensing of the Atmosphere. He has been actively involved in the research and development of weather radar systems for over 30 years. He has played a key role in developing the CSU-CHILL National Radar Facility as one of the most advanced meteorological radar systems available for research and continues to work actively with the CSU-CHILL radar, supporting its research and education mission. He is an avid experimentalist conducting special experiments to collect in situ observations to verify the new techniques and technologies. He has served as an Academic Advisor for over 70 graduate students. He has authored two textbooks and five general books and over 250 peer-reviewed journal articles.*

Step-Down Functions in Airframe Learning Curves: Do They Exist?

Susan L. Moore

Jonathan D. Ritschel, Ph.D.

Edward D. White, Ph.D.

Clay M. Koschnick

Brandon M. Lucas

Shawn M. Valentine

Introduction

Defense cost analysts employ a multitude of techniques to estimate the cost of a weapon system. One of the most widely accepted and utilized techniques is learning curve analysis. Learning curves are traditionally used to estimate recurring costs in a production process (Mislick & Nussbaum, 2015). While previous researchers have studied learning curves along a multitude of dimensions, one area that lacks empirical examination in defense programs is the concept of a step-down function. This lack of empirical examination has led to some debate on whether step-down functions should be employed in cost estimates. We examine the evidence in military fighter airframes to shed light on the issue. Thus, the purpose of this article is four-fold: 1) empirically detect step-down functions in defense aircraft programs 2) examine the impact of weight normalization on step-down functions 3) analyze factors that impact step-down functions and 4) develop an empirically based Cost Estimating Relationship (CER) to predict first unit production costs based upon development unit cost data.

The Step-Down Function

The production of an end item often begins with prototype or development units. This presence of prototype or development units has created the idea of a step-down function. More specifically, step-down functions occur between the Engineering, Manufacturing, and Development (EMD) and the Production phases and are a method for estimating the theoretical first unit production cost based on development asset data (Mislick & Nussbaum, 2015). In learning curves, a step-down would appear as a downward shift on the graph with learning resuming at the same or a modified slope.

The theory undergirding step-down functions is that the development unit is a near production copy in design, physical, and performance characteristics, but is usually accomplished in an EMD environment rather than a production line set-up (Hardin & Nussbaum, 1994). Therefore, the cost to manufacture a development asset is expected to be more expensive than a production model (Hardin & Nussbaum, 1994). Mathematically, the ratio of the production phase first unit cost to the development first unit cost (or average development unit cost) is known as a step-down factor. The actual cost difference between the development first unit cost (or average development unit cost) and the production first unit cost is the step-down (Mislick & Nussbaum, 2015). When first unit production costs are greater than development costs it is called a step-up rather than a step-down.

The Government Accountability Office (GAO) identifies two main learning curve methodologies: Continuous and Step-Down (GAO, 2020). The Step-Down methodology is further broken into two subcategories consisting of Sequential and Disjoint theory. These three models are shown in Figure 1. For ease of visualization, the models are shown in log space.

Continuous learning curve theory is the traditional learning curve described by Wright (1936) for aircraft production but includes the developmental units as part of the curve as shown in Figure 1(a). As such, continuous learning curve theory assumes the same improvement slope in production as well as development. The production estimate can simply be calculated by continuing down the curve for the desired quantity (GAO, 2020).

The two subsets of Step-Down theory, Sequential and Disjoint, typically assume that the improvement slope remains the same in development and production but there is a step down in the value between the cost of the first development unit and the cost of the first production unit (GAO, 2020). Sequential theory states that the cost improvement continues when the first production unit equals the last development unit plus one. For example, if the last development unit is 10, then the first production unit would be 10 + 1 = 11. Initially, Sequential theory sounds like Continuous theory where you consider learning made in development and apply it to production units. Where it differs is that there is a discontinuity in the curve between development and production as shown in Figure 1(b).

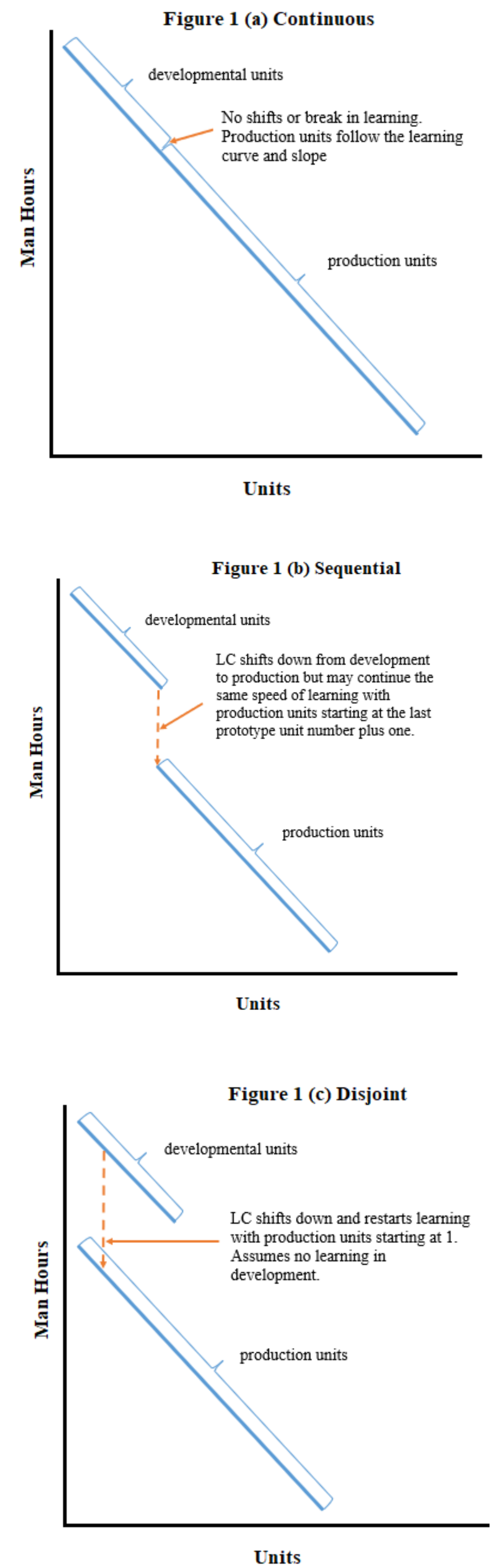


Figure 1. Step-Down Function Types in Log Space

Table 1: Summary of Step-Down Studies

Author	Data	Focus	Method	Conclusion
Waller (1976)	DoD electronics data set (data unavailable)	Formulate step-down factors using disjoint and sequential theories	Compared theoretical first production unit (T1s) developed using disjoint and sequential theories	Mixed results; There was no clear conclusion to whether disjoint or sequential theory was a better predictor for all
Federic (1979)	Same data set as Waller (1976) and included hypothetical data	Cost continuity does not have to be distinctly disjoint or sequential but a potential spectrum between the two theories	Cost improvement curves were fit using different points on the spectrum between production quantities only and prototype and production quantities together	Mixed results; There was no clear conclusion to whether disjoint or sequential theory was a better predictor for all
Hubach, Pehrsson, & Fox (1987)	8 Airframe systems, 6 engines, 6 avionics systems	Determine what is the appropriate range of production data to use in fitting cost improvement curves	Used Ordinary Least Squares (OLS) regression to calculate 6 curves for each system – 3 disjoint and 3 sequential curves.	Mixed results; Airframe did best under disjoint theory. Engine modeling was marginally better using sequential modeling. Avionics had inconclusive results
Malcolm (1991)	7 Marine amphibious assault vehicles	Focused on the relationship between development and production unit costs	Used OLS regression with disjoint and sequential theories	Sequential model was the most applicable to estimating the costs of the amphibious assault vehicles
Hardin & Nussbaum (1994)	Reviewed 9 step-up/ step-down studies	Analyzed the relationship between development and production costs and compared it to other step-down or step-up studies	N/A, no unique model development.	A general step-up or step-down factor can be applied to all types of systems, but the equation would have a much higher variance
Cherwonik et al. (2012)	2 assault vehicles	Use a reference point other than T1 to calculate the learning curve and examined the step-up/down factors	Created basic production learning curves and then calculated step-down factors for prototype to production units	Utilizing a prototype Average Unit Cost (AUC) compared to T1000 provided the least varied step factor
Bui (n.d.)	6 prototype aircraft airframes, 12 production aircraft airframes, and 6 tactical missiles	Analyzed production step-down factors for aircraft and tactical missile manufacturing experiences	Use OLS regression to generate learning curves and determine the step-down percentage from the calculated T1 and average prototype unit cost	Airframe step-down was between prototype and Full Scale Development (FSD) and had lower step-down factors. Missile step downs were higher but occurred between FSD and production

Disjoint theory restarts improvement at the first production unit and does not consider learning created during development phases to be significant (GAO, 2020). Disjoint theory has a curve displacement, but the improvement starts over at unit one rather than at the last development unit plus one as shown in Figure 1 (c). Because it restarts learning, disjoint theory usually results in significantly lower production estimates (GAO, 2020).

Previous Step-Down Function Studies

To the best of our knowledge, there are no previous step-down function studies published in peer-reviewed literature. However, we were able to find seven reports (or conference presentations) specifically related to step-down functions in defense programs. These reports are summarized in Table 1.

There are several key points from the studies in Table 1. First, note that all but one of the step-down studies are more than 25 years old. Second, the majority of the studies have very small sample sizes. For example, the most recent study from 2012 only examined two assault vehicles. Third, many of the studies, in addition to examining the step-down function itself, attempted to develop a Cost Estimating Relationship (CER) between development and first unit production costs. Thus, one of the goals of this article is to develop a new CER with more recent data from fighter aircraft for modern-day practitioner use.

Perhaps the most comprehensive study we discovered is by Hardin and Nussbaum (1994). They reviewed nine internal Navy studies that focused on missile systems, radar, and general electronics. Part of their study examined CERs developed for these disparate system types both individually and as an aggregated CER. Their conclusion was that there could be a general (i.e. aggregated) step-up or step-down factor CER that can be applied to all types of systems, but that equation would have a much higher variance. Therefore, in general, they recommend using system specific step-up/step-down factors in lieu of a CER that applies to all system types. This finding from Hardin and Nussbaum provides the motivation for our study focusing on fighter aircraft programs as a single system type.

Data

The data is primarily sourced from Contractor Cost Data Summary Reports, or DD 1921-2s (Progress Curve Reports), via the Life Cycle Management Center (LCMC) at Wright-Patterson Air Force Base, Ohio. The focus of this study is fighter airframes. The original dataset included 18 programs with 513 lots. The following four criteria had to be met for a program to be included in the final dataset:

1. Have at least one development lot
    a. If no development lot is listed based off DD 1921-2, the early lots can be deemed development if the absolute value of the airframe weight is at least 5% different than its successor *and* if the lot has less than three aircrafts manufactured
2. Have at least four or more production lots
3. Have direct man-hour data for each lot
    a. If less than 20% of direct man-hour production data was missing, the data was imputed
4. Is fighter airframe

The first inclusionary criterion is to ensure that there is adequate data to formulate a step-down factor. To maximize the programs that can be included, two sub-criteria had to be met. Both criteria were developed by reviewing the data available that had development lots identified and by speaking to a subject matter expert at the Air Force Life Cycle Management Center (S. Valentine, personal communication, October 27, 2021). The sub-criteria were purposely made to be conservative in nature to ensure that the inclusion of any lots in the final dataset would not skew the results.

The second inclusionary criterion is to ensure that there was adequate data to calculate the production theoretical first unit (T1). The third inclusionary criterion is that the programs must have complete direct man-hour data for each lot. To maximize the programs that could be included, programs with minimal missing direct man-hour data were reviewed. If the program's production lots were missing less than 20 percent of its direct man-hour data, a line was fitted to the available production lots and an equation was generated. The equation was used to calculate the missing production direct man-hours per aircraft. Table 2 shows the final dataset after employing the inclusion/exclusion criteria.

Table 3 shows the 10 programs analyzed in this article. The program names have been omitted and are designated as Program A, Program B, etc, as a precaution in protecting the data. Note the two programs with asterisks. First, Program F, had missing production man-hour data for two production lots. However, because Program F was missing less than 20 percent of the direct man-hour data, the hours were derived in accordance with inclusion criteria 3(a) above. Second, Program J's development lot was categorized in accordance with inclusion criteria 1(a) above. This program did not have a development lot annotated on the 1921-2 due to its unique acquisition history.

Table 2: Inclusion/exclusion criteria describing the establishment of the final analyzed dataset

	Number of programs	Number of entries
Original dataset	18	513
No development lots	4	69
Less than four production lots	1	6
Missing >20% direct man-hours for lot(s)	2	114
Not airframe	1	12
Remaining dataset	**10**	**312**

Table 3: Final Dataset

Aircraft	Service	Type	Dev Lot(s)	Prod Lot(s)	Total
Program A	Air Force	Fighter	1	8	9
Program B	Navy/ Marine	Fighter	1	50	51
Program C	Air Force	Fighter	1	13	14
Program D	Air Force	Fighter	2	55	57
Program E	Air Force	Fighter	7	33	40
Program F*	Air Force	Fighter	9	11	20
Program G	Navy	Fighter	3	67	70
Program H	Air Force	Fighter	1	12	13
Program I	Marines	Fighter	6	4	10
Program J**	Air Force	Fighter	2	26	28
				Total	**312**

Methods

The term *improvement curve* denotes that costs are used as the primary measure. While there are merits to employing improvement curves, using cost as the dependent variable has some well-documented limitations. These limitations include concerns with wrap rates (Mislick & Nussbaum, 2015), economies of scale (GAO, 2020), and escalation (Hogan et al., 2020). An alternative unit of measure for calculating learning curves is hours and is the approach this article takes. Practitioners have noted that as a program progresses and both cost and hours are provided by contractors, hours are the preferred learning curve unit of measure (S. Valentine, personal communication, August 25, 2021). In addition, negotiations between the government and contractor regarding the program learning curves are typically discussed from a man-hour perspective. These hours are reported on DD 1921-2s. More specifically, the direct man-hours per aircraft from the DD 1921-2 is the summation of four categories: engineering, tooling, quality control and manufacturing. Of these categories, only quality control and manufacturing are accounted for in learning curves and included in our dataset.

Determining Step-Down Factors

The first objective of this article is to determine whether there are step-down factors in the ten fighter aircraft programs. Two unique step-down factors are calculated for each program. The theoretical first production unit (T1) is calculated via OLS regression analysis. This calculated production T1 is then divided by the development first unit direct man-hours (EMD FUH) (note: for purposes of this research this is considered equivalent to development T1). Next, the production T1 is divided by the Engineering and Manufacturing Design (EMD) average unit hours (EMD AUH). The two equations are shown below:

$$\text{Step-Down Factor 1} = \text{Production T1/EMD FUH} \tag{1}$$

$$\text{Step-Down Factor 2} = \text{Production T1/EMD AUH} \tag{2}$$

These step-down factors will then be tested via a Sign test. The Sign test is the non-parametric equivalent of a paired *t* test where it tests for consistent differences of two groups using the median (Shier, 2004). The non-parametric test is required because the total amount of programs reviewed is less than 30 and a particular distribution cannot be assumed.

This Sign test will be conducted for both step-down factor calculations (EMD FUH and EMD AUH). The Sign test is based on the direction of the plus and minus sign of the observation and not on their numerical value. In other words, the Sign test will determine as a group of programs whether the actual first development lot direct man-hours or development average unit hours are statistically different than the calculated production T1. The hypotheses for the Sign test are as follows:

H_0: Difference in median of the signed differences = 0
H_a: Difference in median of the signed differences ≠ 0

Deriving a Cost Estimating Relationship (CER)

Previous step-down studies (Hardin & Nussbaum, 1994) and cost estimation textbooks (Mislick & Nussbaum, 2015) highlight the utility of a CER for practitioner use. The goal of these CERs is to provide a basis for determining an aircraft's production T1 when development data exists. Our first CER uses the development FUH as the independent variable (x) and the calculated production T1 as the dependent variable (y). The data will be fit to a linear (see Equation 3) and power (see Equation 4) function using a non-linear solver; the adjusted R^2 will be used to determine which equation best fits the data. These models will also be used for EMD AUH as the independent variable and the calculated production T1 as the dependent variable. Thus, a total of four regression models will be evaluated.

$$y = \beta_0 + \beta_1 x_1 + \varepsilon \tag{3}$$

Where:
y: Production T1
x_1: EMD FUH or EMD AUH

$$y = Ax^b + \varepsilon \tag{4}$$

Where:
y: Production T1
x: EMD FUH or EMD AUH

Impact of Weight Normalization on Learning Curves

Normalization by weight in learning curves is not a widespread approach. However, some practitioners advocate for it and there is precedence in the literature. For example, Alchian's (1950) study of 22 bomber, fighter, trainer, and transport airframes after World War II normalized the data using direct labor hours per pound. Therefore, we examine the impact of weight normalization in our dataset.

The normalization is accomplished by dividing the program's direct man-hours by the airframe weight. Next, we repeat the Sign tests as described in the *Determining Step-Down Factors* section previously with the newly normalized data. The results from the Sign test will indicate whether a step-down function exists in the data. Comparing the results from the Sign tests of the non-normalized to the normalized data will illuminate any impacts from weight normalization.

Results

We first developed step-down factors for the ten aircraft programs. Recall that two unique step-down factors (see Equations 1 and 2) are calculated for each program: one using EMD FUH and one using EMD AUH. A factor below 1 means that the EMD FUH or EMD AUH had higher direct man-hours per aircraft (step-down). A factor above 1 means that the EMD FUH or EMD AUH had lower direct man-hours per aircraft (step-up). Some programs have the same step-down factor for both EMD FUH and EMD AUH calculations due to only having one development lot. These programs are marked with an asterisk in Table 4.

Table 4: Step-Down Factors

Aircraft	Type	Step-Down EMD FUH	Step-Down EMD AUH
Program A*	Fighter	0.675	0.675
Program B*	Fighter	0.876	0.876
Program C*	Fighter	1.316	1.316
Program D	Fighter	0.631	0.822
Program E	Fighter	0.36	0.726
Program F	Fighter	0.168	0.212
Program G	Fighter	0.376	0.506
Program H*	Fighter	0.833	0.833
Program I	Fighter	0.391	0.493
Program J	Fighter	0.935	1.411
* Denotes only one development lot			

The step-down for the EMD FUH calculation ranged from 0.168 to 1.316. The range for EMD AUH is 0.212 and 1.411. These airframes have a mean of 0.656 and standard deviation of 0.344 for the EMD FUH and 0.787 mean and 0.364 standard deviation for EMD AUH. Only Program C and Program J had step-up factors.

Table 5: Sign Test Results

	EMD FUH	EMD AUH
Test Statistic	22.5	17.5
Prob > |z|	0.0039	0.084

The data in Table 4 appears to show a consistent step-down factor for both calculations. Testing for statistical significance of that observation is discerned by the Sign test. The null hypothesis is that there is no step-down factor, and the alternative hypothesis is that there is a step-down factor. This test uses an alpha of 0.1. Results of the Sign Test is shown in Table 5.

Both EMD FUH and EMD AUH rejected the null hypothesis. This means that, for the sample, there is a statistically significant step-down factor between development and production. These initial results indicate that practitioners developing estimates on fighter airframes should consider incorporating a step-down factor in their estimate.

Predicting Production T1: the Cost Estimating Relationship

Some previous step-down factor research (Malcolm, 1991; Hardin & Nussbaum, 1994) developed CERs for practitioner use in predicting production T1 values from EMD data. Thus, the next step of this research uses the step-down calculations from Table 4 to create CERs. Equation forms were limited to linear and power functions due to their prevalence in learning

Table 6: Cost Estimating Relationship

Step-Down Factor Type	Equation Type	Equation	Adjusted R^2
EMD FUH	Linear	$y = 0.3939x + 27.1994$	0.4983
EMD FUH	Power	$y = 0.0003x^{2.3502}$	0.4246
EMD AUH*	**Linear**	$\boldsymbol{y = 0.5842x + 28.847}$	**0.8149**
EMD AUH	Power	$y = 0.9083x^{0.9618}$	0.8222
*Recommended CER is bolded in the Table			

curve use. A total of four individual equations were created. See Table 6.

As shown in Table 6, we recommend using the EMD AUH linear CER. The adjusted R^2 is nearly equivalent between the linear and power AUH models. However, by choosing the linear CER, OLS regression can be utilized for further evaluation. This practical consideration trumps the minor decrease in adjusted R^2 incurred by selecting the linear model. The recommended CER has a coefficient of variation (CV) of 0.3222. This implies that the CER is a good starting point for a cost analyst to use, but some caution is advised due to the moderately high CV.

Weight Normalized Step-Down Factor Analysis

As previously discussed, weight normalization in learning curves is advocated by some practitioners and is also found in previous literature. Therefore, we normalize the data by weight and recalculated the step-down factors to determine the impacts. Note that one platform, Program G, is excluded from our original dataset due to lack of airframe weight information. The resultant step-downs are shown in Table 7.

The mean and standard deviation for EMD FUH is 0.689 and 0.345, respectively. These values are

Table 7: Step-Down Factors Normalized by Weight

Aircraft	Type	Step-Down EMD FUH	Step-Down EMD AUH
Program A*	Fighter	0.67	0.67
Program B*	Fighter	0.909	0.909
Program C*	Fighter	1.296	1.296
Program D	Fighter	0.631	0.821
Program E	Fighter	0.371	0.749
Program F	Fighter	0.176	0.247
Program H*	Fighter	0.774	0.774
Program I	Fighter	0.401	0.505
Program J	Fighter	0.968	1.462
* Denotes only one development lot			

higher than the mean and standard deviation of the non-normalized data. Similarly, the mean and standard deviation for EMD AUH is 0.826 and 0.371, respectively. These EMD AUH results are also higher than the non-normalized data. These higher mean values indicate that normalizing by weights reduces the impact of a step-down factor. In other words, the reduction in hours for the first unit of production from its prototype development hours is less when normalized for weight than when it is not normalized for weight.

Next, the Sign Test is conducted for the weight normalized data. The alpha is 0.10 and the results of the Sign Test are shown in Table 8.

Table 8: Sign Test Results (Normalized by Weight Data)

	EMD FUH	EMD AUH
Test Statistic	16.5	9.5
Prob > |z|	0.0547	0.3008

The EMD FUH Sign Test rejects the null hypothesis. This suggests that for the sample, there is a statistically significant step-down factor between development and production. This finding is consistent with the finding from the non-normalized EMD FUH data in Table 5. However, EMD AUH fails to reject the null hypothesis of the Sign Test. This finding is contrary to the finding from the non-normalized EMD AUH data in Table 4 which rejected the null. The normalized EMD AUH result indicates that there is *not* a step-down between development and production. In other words, normalizing for weight matters in the EMD AUH calculations.

The contradictory findings of the Sign Test in Table 8 between EMD FUH and EMD AUH warrants further investigation. We hypothesize the difference may lie in “legacy” versus “modern” aircraft. The rationale is that touch labor in legacy aircraft was simpler, with machinists completing fewer complex tasks, in a pre-computer environment. To discern if this is the case, we divide the aircraft into the legacy and modern categories via subject matter expert inputs. Next, we run a Wilcoxon Rank Sum test. The Wilcoxon Rank Sum test is a non-parametric test that tests the locations of each set of data. If the distribution of each dataset is the same, then the location can be interpreted as the median (McDonald, 2014). The Wilcoxon Rank Sum test compares the following hypotheses:

Table 9: Wilcoxon Rank Sum Test Results

	Original Data		**Normalized by Weight**	
	EMD FUH	**EMD AUH**	**EMD FUH**	**EMD AUH**
Test Statistic	2.0254	2.2386	1.8371	2.327
Prob > |z|	0.0428	0.0252	0.0662	0.02

H_0: Median ranks are the same
H_a: Median ranks are different

The null hypothesis states that there is no difference in the step-down factor between the legacy and modern aircraft. The alternative hypothesis states that there is a difference in the step-down factors. The results of the Wilcoxon Rank Sum test are shown in Table 9.

As shown in Table 9, all tests reject the null hypothesis at an alpha level of 0.10. This indicates that there is a difference in the step-down factors across all four measures. These results lend credence to the aforementioned suggestion that the mixed results of Table 8 are likely due to differences in the six modern verses the four legacy aircraft. To corroborate these findings, we rerun the Sign Test previously performed, but this time we *only* include the modern fighter aircraft. The results of this new test rejected the null and supports our hypothesis. However, due to the low *n* value of six associated with just examining the modern fighter aircraft subset, the results of that Sign Test cannot be fully trusted. Therefore, while we mention this robustness check, we caution the reader that this result must be taken with a grain of salt, and therefore we do not show the actual test results.

In summary, there are three key points associated with weight-normalization. First, step-down factors exist even when normalizing by weight. Second, the impact of normalizing by weight, however, is to dampen the magnitude of the step-down factors. Lastly, when calculating weight normalized step-down factors, it is imperative to separate the modern from the legacy aircraft.

Factors that Impact Step-Down Functions

The final analysis examines the impact of factors that the GAO (2020) has identified as important to consider when developing a step-down factor in learning curves. The four factors identified by the GAO are:

A break from the last prototype unit to production
Similarity between prototype units and production units
The production rate
The extent to which the same facilities, processes, and people are used in development and production

Only five of the programs in our study had the requisite data to examine the four GAO criteria. Within those five programs, it was discovered that all five in the sample had similar development and production aircraft and had minimal changes in facilities, processes, and people. Those results effectively removed two of the four GAO considerations (numbers two and four) from the analysis. A regression was conducted with the two remaining factors (number one and three above). It found the production rate to be a significant factor with a *p*-value of 0.0058. The positive sign supports the intuition that the larger the production rate, the bigger the step-down factor will be in the learning curve.

We strongly caution that this result from the GAO influential factors is not conclusive. Our data sample of five was simply too small to draw any definitive conclusions. Additionally, as discussed above, our data only looked at two of the four factors. While we are encouraged that the singular result we found did align with the theory, future research with a more robust dataset is needed to have confidence in the result.

Discussion and Conclusion

The debate regarding step-down factors begins at the most fundamental level regarding whether they exist or not in defense aircraft. Our examination of fighter airframes provides empirical evidence that step-downs are present. In our dataset, the mean step-down factor was found to range from 0.656 (FUH) to 0.787 (AUH), which is a significant reduction in hours for the first unit of production from its development unit.

While we were able to detect the presence of step-down functions, we did not attempt to discern whether the step-down function is Sequential or Disjoint. Some of the prior non-peer reviewed studies from Table 1 attempted to delineate between the two. However, we did not believe the nature of our data lent to such a determination. As a result, the nature of the step-down function (Sequential or Disjoint) remains an open question.

A second issue that is debated is the impact of weight normalization on step-down functions. We find that weight normalization does have an impact in fighter airframes, but it only dampens the magnitude of the step-down rather than removing it fully. The magnitude of the mean differences are approximately 6% for both FUH and AUH calculations. This implies that those practitioners who choose to normalize by weight should show smaller hour reductions. Additionally, it is important to segregate the data between modern and legacy platforms if weight normalization is your preferred approach. Overall, the authors remain agnostic to whether practitioners choose to normalize by weight or not. We simply reiterate the step-downs will still occur in most cases, but to a lesser extent.

Lastly, we provide a recommended CER to estimate the theoretical first unit production cost with development data. Our recommended form is linear with average unit development hours as the independent variable. The simplicity of the CER and ease of implementation mirrors the prior DoD studies (Malcolm, 1991; Hardin and Nussbaum, 1994). Thus, we believe this has great potential for practitioner adoption.

In summary, this article is a significant step forward in understanding step-down functions in DoD programs. With advancement, however, comes limitations that merit acknowledgment. Specifically, the small sample sizes in our tests mutes the statistical results. In some cases, such as the examination of the four postulated GAO factors, the lack of data meant even exploratory examination was not possible. These limitations, however, present an opportunity for future researchers. While we focused solely on fighter aircraft airframes, there is the potential to replicate our analysis with other platform types. Similarly, as more data is collected, a more robust investigation into the factors that impact step-down functions can occur. All these endeavors can add to a fuller understanding of step-down functions in military systems. We hope this article provides a launching ground for these future research efforts.

References

Alchian, A.A. (1950). *Reliability of progress curves in airframe production, (Report Number RM-260-1)*. The RAND Corporation. https://www.rand.org/pubs/research_memoranda/RM260-1.html.

Bui, J. (n.d.). *Analyzing production step down with application to space hardware*. Institute for Defense Analyses.

Cherwonik, J., Craig, C., & Liang, W. (2012). *Learning curve step-down analysis: Abrams main battle tank and Bradley fighting vehicle final report*. Technomics Inc.

Frederic, B. (1979). *Further investigations of the relationship between hardware development costs and hardware production costs in military systems*. Tecolote Research Inc.

Government Accountability Office. (2020). *GAO-20-195G Cost estimating and assessment guide*. https://www.gao.gov/assets/gao-20-195g.pdf

Hardin III, P.L., & Nussbaum, D.A. (1994). *Analyses of the relationship between development and production costs and comparisons with other related step-up/step-down studies*. Naval Center for Cost Analysis.

Hogan, D.W., Elshaw, J.J., Koschnick, C.M., Ritschel, J.D., Badiru, A.B. & Valentine, S.M. (2020). Cost estimating using a new learning curve theory for non-constant production rates. *Forecasting,* 2(4): 429-451. https://doi.org/10.3390/forecast2040023

Hubach, S.O., Pehrsson, K.M., & Fox, T.B. (1987). *Prototype to production step-down model final report*. Management Consulting & Research Inc.

Malcolm, D.S. (1991). *Parametric cost estimation utilizing development-to-production relationship applied to the advanced amphibious assault vehicle*. [Master's Thesis, Naval Postgraduate School]. DTIC https://archive.org/details/DTIC_ADA245920

McClave, J.T., Benson, P.G., & Sincich, T. (2014). *Statistics for business and economics* (12th ed.). Pearson.

McDonald, J.H. (2014). *Handbook of biological statistics* (3rd ed.). Sparky House Publishing. Retrieved November 19, 2021, from http://www.biostathandbook.com/HandbookBioStatThird.pdf.

Mislick, G.K., & Nussbaum, D.A. (2015). *Cost estimation: Methods and tools*. John Wiley & Sons, Inc.

Shier, R. (2004). *Statistics: 2.1 the sign test - www.statstutor.ac.uk*. Statistics: 2.1 The Sign Test. Retrieved November 19, 2021, from https://statstutor.ac.uk/resources/uploaded/signtest.pdf.

Waller, W.E. (1976). *The relationship between development and production hardware costs in military weapons systems*. Tecolote Research, Inc.

Wright, T. P. (1936). Factors affecting the cost of airplanes. *Journal of the Aeronautical Sciences*, 3(4), 122-128. https://doi.org/10.2514/8.155

Captain Susan Moore, *is a cost analyst at the Space Systems Command, Los Angeles AFB, CA. She holds a BS in Business Administration with a concentration in Finance from the University of California, Riverside and a MS in Cost Analysis from the Air Force Institute of Technology (AFIT). Her primary research interests include learning curves, public choice, and cost analysis. (Email address: Susan.Moore.12@us.af.mil)*

Dr. Jonathan D. Ritschel *is an Associate Professor of Cost Analysis in the Department of Systems Engineering and Management at AFIT. He received his BBA in Accountancy from the University of Notre Dame, his MS in Cost Analysis from AFIT, and his PhD in Economics from George Mason University. Dr. Ritschel's research interests include public choice, cost analysis, and economic institutional analysis. (E-mail address: Jonathan.Ritschel@aft.edu)*

Dr. Edward D. White *is a Professor of Statistics in the Department of Mathematics and Statistics at AFIT. He received his BS in Mathematics from the University of Tampa, MAS from The Ohio State University, and PhD in Statistics from Texas A&M University. His primary research interests include statistical modeling, simulation, and data analytics. (E-mail address: Edward.White@aft.edu)*

Lt Col Clay M. Koschnick, PhD*, is an Assistant Professor of Systems Engineering in the Department of Systems Engineering and Management at AFIT. He received his BS in Operations Research from the United States Air Force Academy, his MS in Operations Research from the Georgia Institute of Technology, and his PhD in Industrial and Systems Engineering from the University of Florida. His research interests including engineering economics, decision analysis, and econometrics. (E-mail address: clay.koschnick@afit.edu)*

Dr. Brandon M. Lucas *is an Assistant Professor of Cost Analysis in the Department of Systems Engineering and Management at AFIT. He holds a BA in History from the University of Texas at Austin, a MA in International Relations and ME in Teacher Education from the University of Oklahoma, a MS in Cost Analysis from AFIT, and a PhD in Economics from George Mason University. Dr. Lucas' research interests include profit analysis, cost & economic analyses, and incentive structures. (E-mail address: Brandon.Lucas@aft.edu)*

Mr. Shawn M. Valenti*ne is an Operations Research Analyst at AFLCMC Cost Staff, Wright-Patterson Air Force Base, Ohio. He holds a BS in Actuarial Science from Ohio University and an MS in Financial Economics from Ohio University. He currently acts as the research technical lead at AFLCMC where he engages in Air Force research initiatives related to aircraft. (E-Mail address: shawn.valentine@us.af.mil)*

Are Agile/DevOps Programs Doing Enough Systems Engineering?

Anandi Hira

Abstract: Agile and DevOps methodologies offer efficient processes to deliver high quality products and deploy them to the users quickly. Many commercial organizations have reported large savings in cost and increased productivity from implementing Agile and DevOps methodologies. MITRE completed a qualitative study of the cost impacts as a result of applying Agile methodologies and expected the Systems Engineering, Integration and Test, and Program Management (SEITPM) costs would either remain the same or slightly increase for Agile programs compared to Waterfall programs (Manring, 2016). However, this paper later demonstrates that data from Space Ground systems suggest that the SEITPM costs (as an entity) are approximately 30% lower for Agile/DevOps programs compared to Waterfall programs. In this research study, I analyze whether the difference in SEITPM costs between Agile/DevOps and Waterfall programs is statistically significant by comparing the means and evaluating the statistical significance of including a categorical variable in a regression. The results indicate that the decrease in SEITPM costs for Agile/DevOps programs is statistically significant. Reduced systems engineering could potentially lead to troubles while implementing the architecture/design or in the product quality of the completed system. Some examples of possible troubles are missing requirements, interface, and integration issues with other software and/or hardware modules/components, latent defects in the code, and high defect rates. To understand whether the reduced SEITPM costs has any adverse effects, I also conduct a survey with major industry prime contractors to determine if their observations reflect Space Ground systems data, what caused the reduction in SEITPM costs, and if they noticed any positive or negative changes in product quality as an effect. In general, organizations have experienced changes in SEITPM activities but have not experienced adverse effects in product quality as a result. Fortunately, Agile and DevOps methodologies provide a way to reduce costs without negative effects on the product's quality.

The main tenets of the Agile methodology include: incrementally gathering requirements, designing the system, developing and testing the code, demonstrating to users to get feedback, and incorporating changes to the requirements and working software. The DevOps methodology encourages faster software development and release to users by putting the development and operations related activities in parallel with each other and automating as much of the process as possible. Traditionally, software was built with sequential steps, using what is called the Waterfall model: first, the requirements were gathered, then the system was designed, after which the developers implemented the system, testers then tested it, and the system was delivered to users and customers upon completion. Following the Agile and DevOps methodologies allow the development team to provide working software quickly by continually demonstrating working features, as well as get guidance on how much to do or when to stop if schedule and budget constraints are reached. Theoretically, the biggest savings were expected in software development and sustainment efforts. MITRE presented the expected cost impacts of applying Agile methodologies, which states that in the best-case scenario, some savings are expected in software development effort and significant savings expected in sustainment effort (see Figure 1) (Manring, 2016).

Life Cycle Cost Element	Cost Impact Range	
	Best Case	Worst Case
Program Management/System Engineering	=	+
Software Development	-	=
Integration and Test	=	+
Fielding/Deployment	=	++
Training	+	++
Sustainment	--	-

++ significance increase, + increase, = no impact, - decrease, -- significant decrease

Figure 1. Recreation of MITRE's image demonstrating cost impacts of Agile methodology on various Cost Elements (Manring, 2016)

SEITPM is an abbreviation that Department of Defense (DoD) programs use to signify the effort and costs spent in Systems Engineering (requirements gathering, architecting and designing of the program), Integration and Test, and Program Management. In the Waterfall software development lifecycle model, the steps of developing a software project (requirements gathering, architecting and designing, coding, testing, and deploying) are followed sequentially. Due to this, there is typically a high level of SEITPM effort and costs that occur at the beginning of a program (primarily due to systems engineering and program management), which quickly drops and levels until the end of the program (for program management), ending in an increase for integration and testing efforts. For Agile/DevOps programs, on the other hand, these SEITPM-type activities (as well software development) are expected to occur at a more constant rate throughout the software development lifecycle. See Figure 2 to visually see the difference of how SEITPM costs are expected to behave differently through a software development lifecycle for Waterfall and Agile/DevOps programs/projects.

Traditionally, different teams were responsible for Systems Engineering, Program Management, and Integration and Test activities. These labor categories were typically considered to be separate from the development activities, and therefore, tracked separately from the development activities. The Agile and DevOps methodologies, however, increase the speed at which requirements can change and those changes can be made in the resulting code by tightly knitting all the activities with the software development efforts (Seaver, 2018).

The definitions of the Waterfall, Agile, and DevOps lifecycle models describe how SEITPM costs theoretically are distributed across the lifecycle. The MITRE study (see Figure 1) suggests that the total SEITPM costs will be the same or higher for Agile programs, but that hypothesis is not based on an empirical analysis (Manring, 2016). This research study will determine whether total SEITPM costs differ between Agile/DevOps and Waterfall programs as the MITRE study suggests. Additionally, I survey several Agile/DevOps teams in industry to understand whether they noticed a change in the

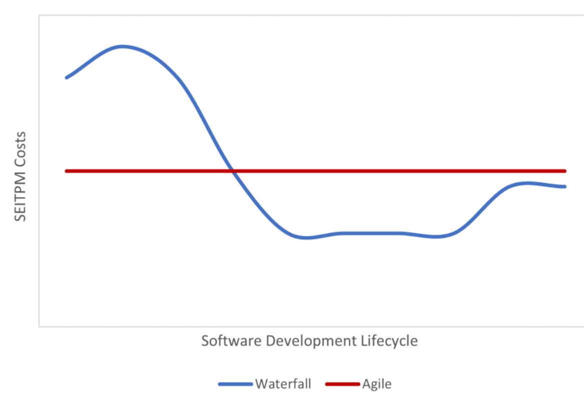


Figure 2. Visual representation of how Systems Engineering, Program Management, and Integration and Test (SEITPM) costs behave through a software development lifecycle for Waterfall and Agile/DevOps programs. This graph is created to visually depict how the costs theoretically differ and is not based on real data.

SEITPM effort/costs, as well as the causes and effects of the changes they observe within the development environments. After a brief introduction to the different software development lifecycle models (Waterfall, Agile, and DevOps), this paper has 2 parts:

1. Empirical comparison of SEITPM costs between Agile/DevOps and Waterfall programs
2. Completed surveys and discussions with Agile/DevOps teams in industry.

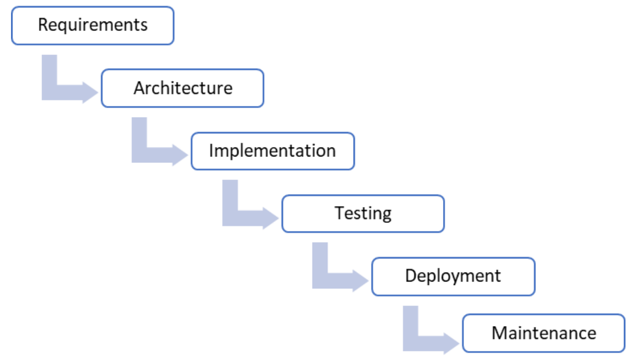


Figure 3. Waterfall software development lifecycle model

Software Development Lifecycle Models

Waterfall

Traditionally, software was developed in sequential steps, as demonstrated in Figure 3. First, the team needs to understand and gather the requirements of what the software system needs to do, then design the system so that the requirements can be satisfactorily met. Taking the completed design and architecture, developers implement the system, followed by testing to ensure that the software works as intended. Finally, the software system is deployed to the users, and maintained as required. The main concept is that each of the steps must be done sequentially in order to fully understand and implement the system correctly.

Software systems had a reputation for high failure rates, budget, and schedule overruns, and not meeting the users' needs. The source of these problems was that working software is only produced at the very end of the waterfall development lifecycle. This caused high amounts of risk and uncertainty in understanding whether the requirements could successfully be met, as well as whether the users would be satisfied with system (Ben-Zahia & Jaluta, 2014). Additionally, it was difficult to assess progress, and testing efforts would often be cut short due to schedule and budget overruns (Davis, 2000). As technology began to change quickly, the completed systems would either no longer be applicable to the current needs or compatible with updated or changed platforms (Sinha & Das, 2021).

Agile

To react to the increasing changes in technology and users' needs, a group of software developers came up with a way to speed up software development and deploy more quickly to market/field. The group developed a manifesto and 12 principles to define the goal and main tenets build software successfully (Beedle, et al., 2001). The main tenets are to shorten the time it takes to get working software to users, and continuously and quickly get feedback from users. The lifecycle model constructed to fulfill the manifesto and the 12 principles are visually described in Figure 4. Instead of performing the steps needed to develop

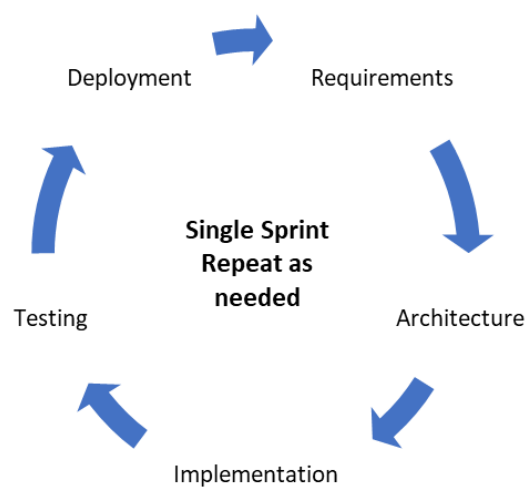


Figure 4. Agile software development lifecycle model

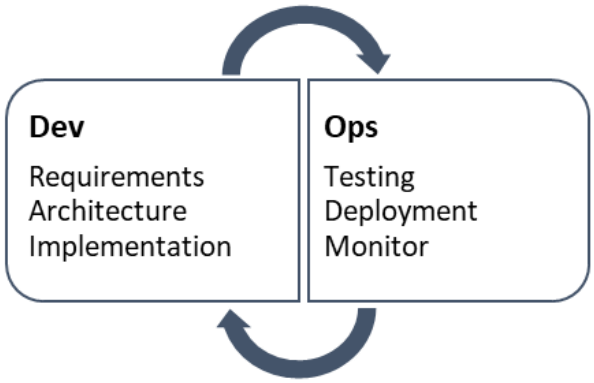


Figure 5. DevOps software development lifecycle model

software sequentially as in the Waterfall model, they are performed iteratively in short "sprints" or iterations throughout the lifecycle. This allows the developers to get feedback from users on a regular basis, demonstrate progress by demonstrating working software, and incorporate changes to the requirements or needs. Many commercial organizations and teams reported being able to deploy software to the market/field earlier, higher development productivity, cost savings, and better customer experience and satisfaction as a result of implementing Agile practices and methodologies (Russo, 2021).

DevOps

While Agile made developing, testing, and deploying software rapidly a common phenomenon, many organizations had separate development and testing teams in order for the testing and verification to be independent from the development efforts. Additionally, many tools to automate various activities (such as developing, testing, and deploying software) became more widely available and highly utilized in development environments. The use of parallel teams and increased use of automation coined the term DevOps to further shorten the development cycle and get operational software out to the users at a faster pace (see Figure 5). Generally, people have been using Agile and DevOps methodologies in conjunction. In some ways,

Dataset	Program Level	Data Description	Data attributes	Data Filters
Dataset A	Total or by Increment	· Targeted Ground systems and software-intensive programs across the Air Force and Space Force. · Data comes from Earned Value reports from contractors, which includes all costs to-date by WBS element. Also includes an Estimate At Completion (EAC) for incomplete programs.	Costs by major program elements (SEITPM, Software, Hardware, and Space segment) as well as software development hours, ESLOC (Equivalent Source Lines of Code), Requirements, Agile-like development process, % Complete, data sources, period of performance in months.	At least 85% complete, to ensure confidence in actual and estimated costs. Also, removed programs included in below dataset.
Dataset B	Annual – summed for Total or Total to Date	· Targeted Ground systems and software-intensive programs across the Air Force and Space Force. · Data comes from the Government's budgeting tool called CcaRs. Based on Contract Line Item Numbers (CLINs).	Costs by major program elements (SEITPM, Software development, and Platform development), as well as ESLOC, Requirements, User Stories, or Story Points.	Programs for which costs could be retrieved to be consistent with the above dataset.

Table 1. Brief description of datasets used

DevOps can be considered as an extension or special case of Agile.

Part 1: Data Analysis

Research Methodology

Datasets

This study uses two sets of data collected from government-funded software development programs collected by the Air Force Cost Analysis Agency (AFCAA), further described in Table 1.

Dataset A focused on identifying whether programs were Agile-like, while Dataset B collected data that followed DevOps processes. Three programs were in both datasets. To avoid double counting these programs, the versions from Dataset A were removed from this analysis. Note, though previous research found that 92.5% complete is equivalent to a complete program (Tracy & White, 2011), this study uses a 85% completion as the threshold to balance between accuracy and retaining data points. Most data points represent large, in-progress programs.

As mentioned in the Software Development Lifecycle Models section above, many teams and organizations utilize both Agile and DevOps processes in conjunction. Therefore, the Agile-like and DevOps programs are grouped together.

Group	Data Sources	# of data points
Waterfall	· Dataset A	30
Agile/DevOps	· Dataset A · Dataset B	27

Table 2 Software Size Metrics

Table 2 shows that there are a comparable number of data points in the 2 groups used in this study.

Base Year Normalization

As mentioned in Table 1, both datasets used in this study provide the costs of major program elements, and these costs are in terms of Then Year dollars (the cost at the time of spending). To ensure that the data and costs are comparable, the costs were normalized to Base Year 2020 (BY20) dollars. The steps to perform the conversions (explained in Table 3) differ by dataset because of how differently the data was collected for both datasets.

SEITPM Estimation Methodologies

Typically, SEITPM effort and costs are estimated in comparison to the Prime Mission Product (PMP), which is the actual software development and infrastructure costs (costs needed to support the development and/or operations environment,

Dataset	Data Source Description	BY20 Conversion Method
Dataset A	Data comes from Earned Value reports, which means the dollars are a cumulative sum of Then Year dollars (dollars' value at time of spending).	***Mid-Point Method*** The mid-point or middle year of a program is used (start and end years are provided in the data) as the original Constant Year (CY), which is then converted to BY20 by applying appropriate escalation indices.
Dataset B	Data comes from budget tool that stores costs on annual basis (in Then Year dollars).	***Sum of Annual Escalations*** Since costs are provided on annual basis, each year's costs are escalated to BY20 dollars. All the converted years' costs of a program are summed up for the total cost.

Table 3. Ways to group and estimate SE, IT, and PM efforts and costs

such as software licenses and supporting hardware). SEITPM is compared to PMP in 2 ways, typically (Markman, Ritschel, & White, 2021):

1. SEITPM is estimated as a factor of, or in proportion to, the PMP costs (SEITPM/PMP)
2. Using a regression where SEITPM costs is a dependent variable and PMP costs is the independent variable. The resulting regression is also called a Cost Estimating Relationship (CER)

As explained previously, SEITPM consists of 3 types of labor/activities: Systems Engineering, Integration and Test, and Program Management. Depending on how teams actually track and bucket their costs and efforts across these 3 activities, it is very common for these 3 activities to be grouped or separated in the 3 ways demonstrated in Table 4.

Analysis Method

The primary objective of this research study is to determine whether there is a significant difference in SEITPM costs between Agile/DevOps and Waterfall programs. As mentioned in the previous subsection, SEITPM is estimated in 2 ways: as a factor of PMP costs or using a regression against PMP costs. Therefore, this study analyzes if there is a difference in SEITPM costs across the Agile/DevOps and Waterfall groups by looking at the data in both ways. A high -level description of the analysis method by type is explained in Table 5.

Numerator or Dependent Variable	**Denominator or Independent Variable**	**Total Costs**
SE + IT + PM	PMP	SEITPM + PMP
SE + PM	PMP	SEPM + PMP **+ IT***
SE + PM	PMP + IT	SEPM + (PMP + IT)

* Note, IT costs need to be added separately to get the total program's cost in the 2nd option/row

Table 4. Ways to group and estimate SE, IT, and PM efforts and costs

Estimation Method	Analysis Method
SEITPM/ PMP Proportion	Compare the means of the SEITPM/PMP proportions, as well as the individual activities' proportions (Systems Engineering (SE), Program Management (PM), and Integration and Test (IT)), between the 2 groups using t-test. The t-test should return a p-value of less than 0.05 for difference to be considered statistically significant. The variables used as inputs are log-transformed and tested for normal distribution using the Shapiro-Wilk test (need a p-value of at least 0.05). If the variables are not normally distributed, the Mann-Whitney test is run, which also requires a p-value
SEITPM vs PMP Regression /CER	Include a categorical/dummy variable for Agile/DevOps and evaluate the p-value of the coefficient, as well as goodness of fit and prediction accuracy statistics. The p-value of the coefficient should be less than 0.05 for statistical significance.

Table 5. Summary of analysis methods by the 2 SEITPM estimation methods

Also explained in the previous subsection are the 3 variants of the SEITPM and PMP costs, and all 3 variants are used in the comparison between Agile/DevOps and Waterfall programs.

Results

SEITPM Proportion Comparison

The t-test is a parametric test, which means that the test assumes the variables used as inputs are normally distributed. Table 6 has the Shapiro-Wilk test p-values for the log-transformed variables (most variables were not normally distributed before the transformation) across the 2 groups (Waterfall and Agile/DevOps), and p-values larger than 0.05 imply the variable cannot reject the null hypothesis of not being normally distributed.

For the variables that returned p-values of less than 0.05 (dark red text in Table 6), the non-parametric Mann-Whitney test is run instead of

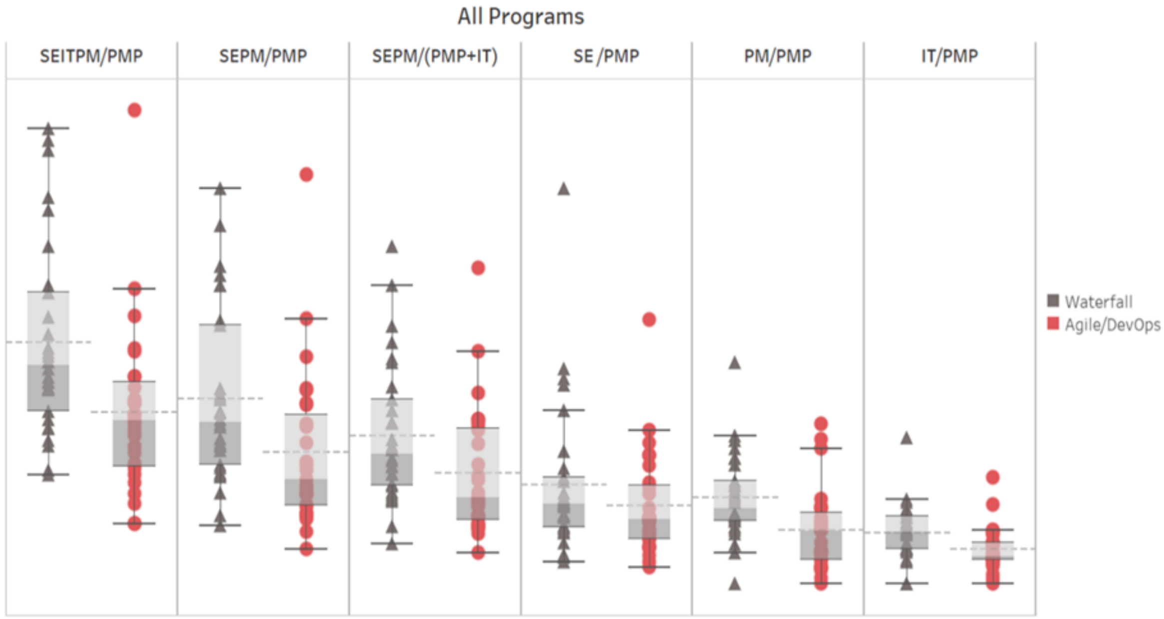


Figure 6. Box plots of the 3 SEITPM variants' and individual activities' (SE, PM, and IT) proportions to PMP costs across Waterfall and Agile/DevOps groups

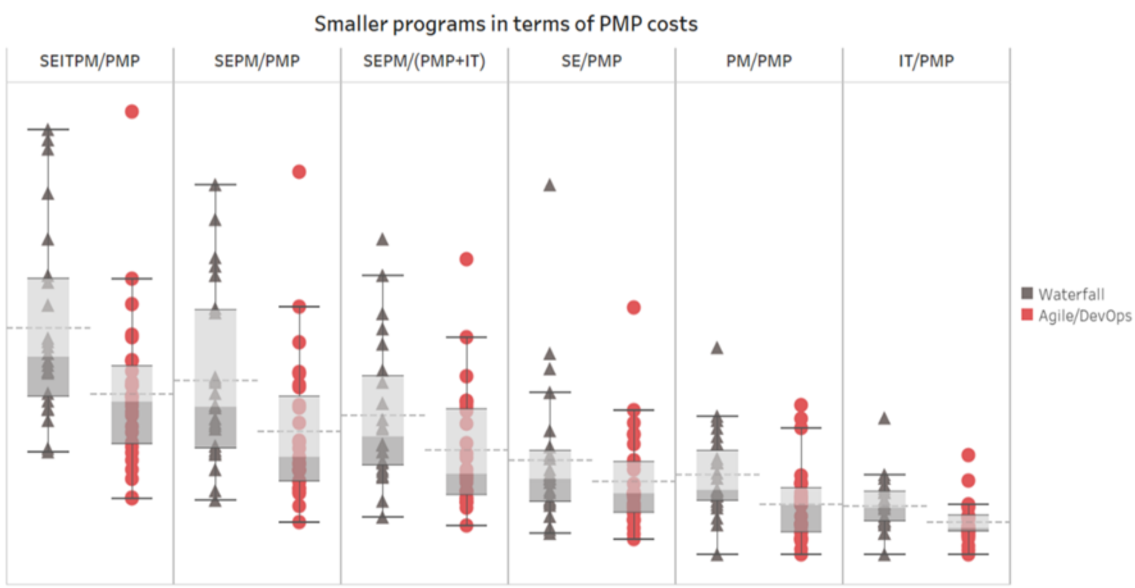


Figure 7. Box plots of the 3 SEITPM variants' and individual activities' (SE, PM, and IT) proportions to PMP costs across Waterfall and Agile/DevOps groups using the subset of smaller programs

	Shapiro-Wilk p-values	
	Waterfall	**Agile/ DevOps**
log(SEITPM/PMP)	0.45	0.81
log(SEPM/PMP)	0.97	0.98
log(SEPM/(PMP + IT))	0.92	0.64
log(SE/PMP)	0.0006	0.02
log(PM/PMP)	0.44	0.06
log(IT/PMP)	0.19	0.04

Table 6. Shapiro-Wilk test for normality p-values on log-transformed variables

	t-test/Mann-Whitney test	
	t-values	**p-values**
log(SEITPM/PMP)	3.295	0.0009
log(SEPM/PMP)	2.84	0.003
log(SEPM/(PMP + IT))	2.13	0.02
log(SE/PMP)		0.08
log(PM/PMP)	3.09	0.002
log(IT/PMP)		0.004

Table 7. T-test and Mann-Whitney test results on log-transformed variables

the t-test. The Mann-Whitney test also needs to return p-values of less than 0.05 for the difference between the groups to be considered statistically significant.

Along with the p-value, the t-test reports a t-value which represents the ratio of the difference between the two groups' means. Therefore, t-values larger than 1 and p-values of less than 0.05 indicate there is a statistically significant difference between the Waterfall and Agile/ DevOps means for the variable being tested. The Mann-Whitney also produces a W-value, but it is the sum of the ranks of the first sample and does not indicate a difference between the 2 samples. The W-value does not provide a sense of difference or proportions between the 2 samples and, therefore, is not reported in this paper.

Table 7 shows the tests' results comparing Waterfall and Agile/DevOps groups and Figure 6 visually demonstrates the differences between the groups using box plots (the proportions on the y-axis are not shown to maintain confidentiality). Both show that SEITPM, PM, and IT proportions of Agile/ DevOps programs are significantly lower than Waterfall programs.

The largest Agile/DevOps program is significantly smaller than several programs in the Waterfall group (in terms of PMP BY\$M). To compare the means of the SEITPM proportions of PMP costs across similarly-sized programs, the dataset is trimmed at programs with PMP costs that are no larger than 5% more than the largest Agile/ DevOps program.

Re-running the above-explained analyses for the smaller programs subset of the data led to the same conclusions: SEITPM, PM, and IT proportions for Agile/DevOps programs are significantly lower than Waterfall programs. SE is the only activity whose difference between the Agile/DevOps and Waterfall groups is not statistically significant. Table 8 and Figure 7 show the statistical test results and the visual representation of the groups' behaviors across the SEITPM variants and individual activities, respectively. As before, the dark red text in Table 8 represents tests with p-values that suggest the

	Shapiro-Wilk p-values		***t-test/Mann-Whitney test***	
	Waterfall	**Agile/ DevOps**	**t-values**	**p-values**
log(SEITPM/PMP)	0.43	0.81	3.06	0.002
log(SEPM/PMP)	0.72	0.98	2.54	0.007
log(SEPM/(PMP + IT))	0.92	0.64	2.13	0.02
log(SE/PMP)	0.004	0.02		0.11
log(PM/PMP)	0.61	0.06	2.69	0.005
log(IT/PMP)	0.12	0.04		0.004

Table 8. Shapiro Wilk and either t-test or Mann-Whitney test results on log-transformed variables across the subset of smaller programs

data cannot be considered normally distributed or that the difference between the groups is not considered statistically significant. The proportions on the y-axis in Figure 7 are not shown to maintain confidentiality, but the SEITPM, PM, and IT proportions are about 30% lower for the Agile/DevOps programs.

SEITPM CER (Cost Estimation Relationship)

In order to rigorously compare and evaluate the regressions' and goodness-of-fit statistics, as well as use a curve that fits the actual trend of how SEITPM costs grow, I log-transformed the variables and ran linear regressions. The 2 regressions I compare are:

1. SEITPM vs PMP without any other variables
2. SEITPM vs PMP with Agile/DevOps categorical variable (set to 1 if the program is an Agile/DevOps program or 0 otherwise)

In both cases, the SEITPM and PMP variables are log-transformed. The Agile/DevOps variable is not log-transformed, and Equation 1 displays how the linear regression is run and how it converts back to unit-space. Therefore, all regression statistics displayed in this section are in log-space, not unit-space. To reduce bias in the regression, I used the Minimum-Unbiased-Percentage Error (MUPE) with Modified Marquardt method, which weighs the data points such that the average error percentage is 0 (Hu, 2001).

$$\log(SEITPM) = a + b \times \log(PMP) + Agile/DevOps \times c$$

$$SEITPM = a \times PMP^{b} \times (10^{c})^{Agile/DevOps}$$

Equation 1 Log-transformed linear regression and conversion to unit-space with Agile/DevOps categorical variable

Figure 8 displays that the trendlines of SEITPM costs against PMP costs for Agile/DevOps programs are, with a few exceptions, consistently and proportionately lower than Waterfall programs. Similar trends are visible when SEPM is graphed against PMP and PMP+IT.

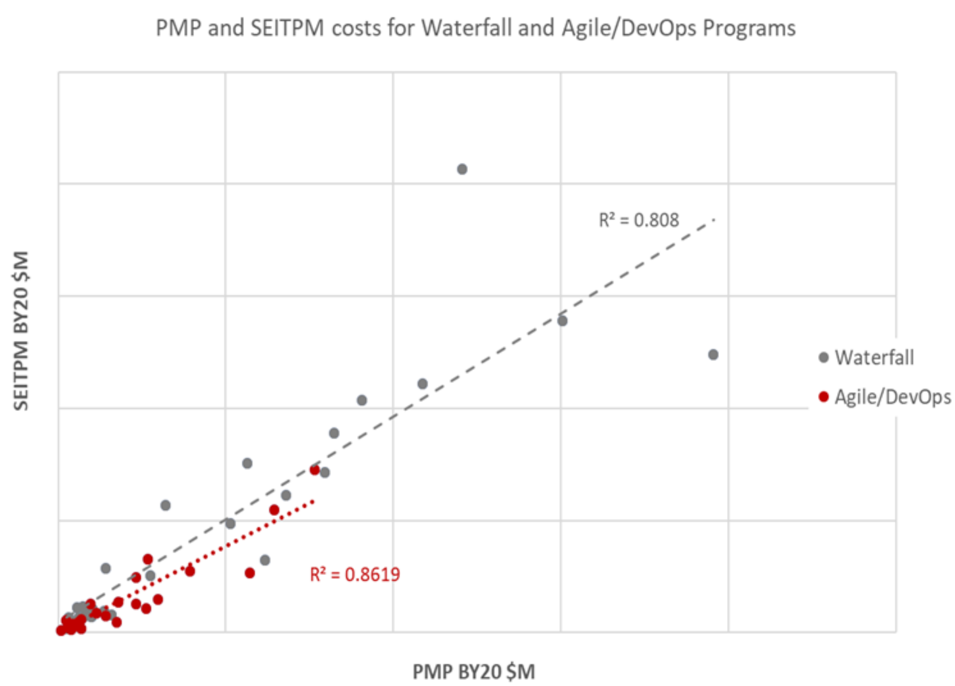


Figure 8. SEITPM costs against PMP (Prime Mission Product) costs trendlines, grouped by development type (Waterfall and Agile/DevOps). Actual data points are removed to preserve the confidentiality of the programs

	Without Agile/DevOps variable			With Agile/DevOps variable		
	SEITPM vs PMP	**SEPM vs PMP**	**SEPM vs PMP+IT**	**SEITPM vs PMP**	**SEPM vs PMP**	**SEPM vs PMP+IT**
Intercept p-value	0.5642	0.6672	0.2724	0.0399	0.3485	0.8554
Agile/DevOps p-value				**0.0028**	**0.0123**	**0.0293**
Adj R^2 for MUPE	85.16%	80.19%	80.62%	87.01%	81.83%	81.79%
Standard Error	0.2026	0.2388	0.2352	0.1849	0.225	0.2261
Average Error %	39.09%	48.78%	47.94%	34.41%	43.95%	44.77%
% of Predictions within 25% of actuals	50.88%	31.58%	31.58%	49.12%	36.84%	33.33%
% of Predictions within 30% of actuals	54.39%	40.35%	42.11%	56.14%	42.11%	43.86%

Table 9. Goodness-of-fit and prediction accuracy statistics for SEITPM/SEPM Regressions/CERs (Equation 2)

$$SETIPM = 1.6734 \times \mathbf{0.6864}^{Agile/DevOps} \times PMP^{0.9005}$$

$$SEPM = 1.3201 \times \mathbf{0.6863}^{Agile/DevOps} \times PMP^{0.8858}$$

$$SEPM = 1.0578 \times \mathbf{0.7206}^{Agile/DevOps} \times (PMP + IT)^{0.8908}$$

Equation 2 SEITPM/SEPM Variants' Regressions/CERs

The p-values on the intercept variable (in log-space) and on the Agile/DevOps variable, and a couple goodness-of-fit statistics on the regressions are listed in Table 9. The 6 regressions are for the 3 variants of SEITPM with and without the Agile/DevOps categorical variable. The results in Table 9 show that Agile/DevOps categorical variable is statistically significant (the p-values are well below 0.05 for all 3 variants of SEITPM) and the goodness-of-fit statistics are better than the regressions without the categorical variable. Additionally, the base/coefficient values for the Agile/DevOps variables (in Equation 2) suggest that SEITPM costs are about 30% lower for Agile/DevOps programs compared to Waterfall programs (similar to the results found when comparing the means in the SEITPM Proportion Comparison subsection above). Note, the resulting regressions/CERs in Equation 2 should not be used without understanding the underlying data and its ranges or for application types or domains not represented in the datasets used in this study.

Conclusion

Analyzing the data available on the Space Ground systems concludes that the SEITPM costs are about 30% lower for Agile/DevOps programs compared to Waterfall programs. Looking at each of the activities separately (SE, PM, and IT), Program Management (PM) and Integration and Test (IT) costs are also significantly lower for the Agile/DevOps programs compared to Waterfall programs. While there is a slight reduction in Systems Engineering (SE) for Agile/DevOps programs, the difference is not considered statistically significant.

These differences can be caused by the differences in the Agile and DevOps methodologies compared to Waterfall, such as:

- Systems Engineering (SE) and Integration and Test (IT) activities should be more incremental and level-loaded, along with software development activities (Seaver, 2018).
- The Agile principles encourages teams to be self-organizing and be part of the task management and decision-making process. Therefore, moving some of the Program Management and Systems Engineering

activities down to the software development team (Beedle, et al., 2001).

- Agile teams are cross-functional and breaking out the effort and costs for specific activities becomes difficult, if not impossible (for example, software development and integration activities) (Beedle, et al., 2001).
- The Agile principles suggest maximizing the amount of work that is not done or streamlining the processes to focus on doing just enough work (Beedle, et al., 2001).

A concern of the reduced activities (and as a result, cost) is whether there would be adverse effects on the product's quality, such as not being able to meet scalability or level of service requirements. To understand whether applying the Agile and DevOps methodologies lead to a reduction in SEITPM costs and whether this reduction leads to lower product quality, the next step of this research was to survey and have discussions with industry partners asking for insights, causes, and effects of the phenomenon.

Part 2: Survey Industry Research Methodology

Survey Questions

The goals of surveying industry were to understand whether or not the software development teams were actively noticing that the Agile/DevOps programs required less SEITPM activities, as well as the causes and effects of this phenomenon. The questions formulated to meet these goals, along with Agile principles or beliefs that support the questions are in Table 10.

Survey Participants

I worked with Space Systems Command (SSC) Financial Management Cost Research (FMCR) department to set up meetings with their industry partners to brief the data analytics results and get their answers on the questions listed in the previous subsection. These industry partners are also represented in the dataset used in the first part of this research study. The suggestions I made for the participants to attend the meeting and respond to the questions were Program Managers, cost analysts, and/or team members that have:

- An understanding of the SEITPM efforts, staffing, and/or costs
- Worked on an Agile/DevOps program that is at least 75% complete
- And also worked on a Waterfall program to be able to comment on the differences between Waterfall and Agile/DevOps programs (or members from both types of programs could also join for real-time comparisons)

The participants were given 2 options for how to order the briefing of the results and answering the questions:

1. Participants could provide responses before the briefing. I would then review the responses and ask follow-up questions after briefing the results.
2. Participants can first view the briefing of the results and dynamically answer the questions during the meeting. This option allowed for participants to get necessary context and background for the questions, which may help participants get clarification and figure out who can answer the questions.

I received responses and held meetings with 5 organizations, using a combination of the two methods above with a combination of Program Managers, cost analysts, and software developers. The organizations and respondents are not mentioned in this paper to maintain confidentiality.

Results

In many cases, the industry partners provided very extensive responses to the questions. In this paper, I provide a summary of the responses that sufficiently answer the questions.

Question 1: Include SEITPM in Scrum/Development Teams?

***Agile Principles** [4]*	***Q#***	***Questions***
Teams should be highly collaborative, self-organizing, and cross-functional.	1	On Agile/DevOps programs, do you include SEITPM FTEs in the Scrum/ development teams? Does the role of the SEITPM FTEs in the Scrum/ Development teams focus only on the Scrum team product? Are any overarching system-level Systems Engineering or system architecture efforts included?
Incrementally gather requirements, develop and test software, and deliver to users.	2	Are SEITPM hours/cost level-loaded across the lifecycle versus high in the beginning for Agile/DevOps programs?
		The data we have suggests that the overarching SEITPM is about 20% lower for Agile/DevOps programs compared to Traditional programs. By looking at each of the activities (Systems Engineering, Program Management, and Integration & Test) separately:
The best architectures, requirements, and designs emerge from self-organizing teams.	3	Systems Engineering may have reduced slightly, but not significantly. Are you noticing if the overarching system-level Systems Engineering is about the same across Waterfall and Agile/DevOps programs? If different, how so and why?
Teams should be highly collaborative, self-organizing, and cross-functional.	4	Program Management is significantly less for Agile/DevOps compared to Waterfall programs. Are you noticing the same behavior? What is causing that (examples: reduced deliverables, management activities being moved into development teams)?
Incrementally gather requirements, develop and test software, and deliver to users.	5	Integration & Test is significantly less for Agile/DevOps compared to Waterfall programs. Are you noticing the same behavior? What is causing that (example: integration and testing efforts being captured within development efforts, as they moved into Scrum/development teams)?
Teams should be highly collaborative, self-organizing, and cross-functional.	6	On 2 different datasets, Causal Inference algorithms found a causal link between analyst and programmer capability. From my previous experiences, I found that teams that had good analytical skills also had the tendency to be better programmers. Have you noticed if the analytical and/or programming skills of the developers improved with SE and PM FTEs being involved in the sprints/iterations?
The best architectures, requirements, and designs emerge from self-organizing teams.	7	Has including SEITPM FTEs in the Scrum/development teams led to improved requirements gathering and accuracy, architectures, and designs?
Incrementally gather requirements, develop and test software, and deliver to users. Incremental deliveries, feedback loops, and frequently tested software lead to better working software and higher customer satisfaction.		Since requirements are gathered and the design/architecture is built incrementally:
	8	Have you noticed positive or negative changes in the quality of products?
	9	Have you noticed any trouble with meeting level of service requirements later in the development lifecycle, compared to when using the Waterfall lifecycle mode?
	10	Has the maintainability of the product improved/decreased for Agile/ DevOps programs compared to Waterfall?
	11	Have you noticed reduction/increase in rework, scrapped code, and defects?

Table 10. Industry Survey Questions

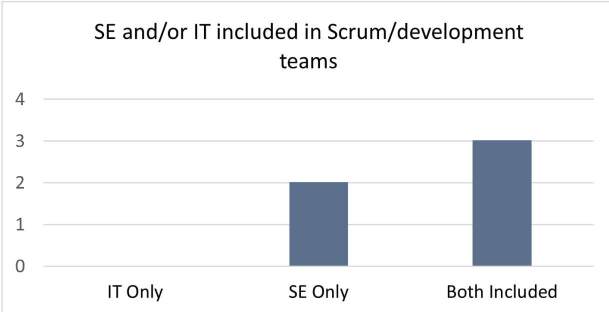


Figure 9. Quantitative Summary of Survey Question 1 Responses

Question: On Agile/DevOps programs, do you include SEITPM FTEs in the Scrum/development teams? Does the role of the SEITPM FTEs in the Scrum/Development teams focus only on the Scrum team product? Are any overarching system -level Systems Engineering or system architecture efforts included?

The goal of the first question is to see if organizations are creating cross-functional teams in practice, and whether systems engineers perform any overarching system-level functions within that role. While systems engineers are needed to ensure that a single component works as expected, Systems Engineering (SE) at the overarching system-level ensures that components are able to integrate and that the system as a whole works as expected. Summaries of responses received are represented in Figure 9.

In general, the industry is creating cross-functional teams that include software developers, systems engineers, and in some cases, testers. However, the SEs typically only serve to provide support in the development of the team's tasks. Hence, no Systems Engineering (SE) that could be attributed to the systems-level is being done within the development/Scrum teams.

Question 2: Is SEITPM level-loaded?

Question: Are SEITPM hours/cost level-loaded across the lifecycle versus high in the beginning for Agile/DevOps programs?

Since Agile and DevOps methodologies promote performing all activities in an iterative fashion, the SEITPM activities and efforts should be mostly level-loaded across the lifecycle in comparison to the Waterfall programs. All industry partners confirmed noticing the same phenomenon.

Question 3: Reduction in Systems Engineering?

Question: Systems Engineering may have reduced slightly, but not significantly. Are you noticing if the overarching system-level Systems Engineering is about the same across Waterfall and Agile/DevOps programs? If different, how so and why?

In the first part of this research study, the Mann-Whitney test suggested the means of SE/PMP were not significantly different between Agile/DevOps and Waterfall programs. With this question, the industry partners let us know whether they noticed any significant reductions in the amount of SE used or needed for Agile/DevOps programs compared to Waterfall ones.

Organization 4 worked on a program where they initially thought they were realizing a 65%

Industry Partner	**Summarized Answer**
Organization 1	Similar amount of SE activities. Maybe some more upfront activities, but balances with savings by including SE FTEs with the development team.
Organization 2	Don't have data, but probably similar between Agile and Waterfall
Organization 3	Slight reduction, but similar. Developers tend to pick up some of the functionality along the way.
Organization 4	Not sure.
Organization 5	One program noticed higher SE activities and costs compared to a typical Waterfall program, but noted that the nature of the program warrants this. On another program, the team is noticing significantly lower SE costs because the activities are being pushed down to the software development teams.

Table 11. Survey Question 3 Response Summaries

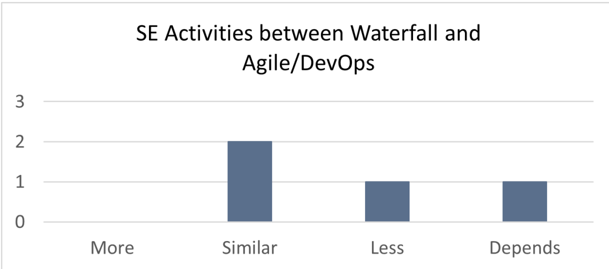


Figure 10. Quantitative Summary of Survey Question 3 Responses

savings in SE costs. Later, they realized they did not do enough SE activities upfront, which led to increased costs later in the lifecycle. Therefore, they are not sure if the SE costs would be lower for Agile/DevOps programs in an ideal scenario. This experience demonstrates a concern that insufficient systems engineering can lead to adverse effects on the program.

In general, the industry partners did not have or analyze their data for whether or not the SE costs were different between Agile/DevOps and Waterfall programs. However, most responses indicate that the team did not notice significant changes in SE activities between Agile/DevOps and Waterfall programs. This may indicate that the industry partners are also being cautious with ensuring that enough systems engineering activities are being done on programs.

Question 4: Reduction in Program Management?

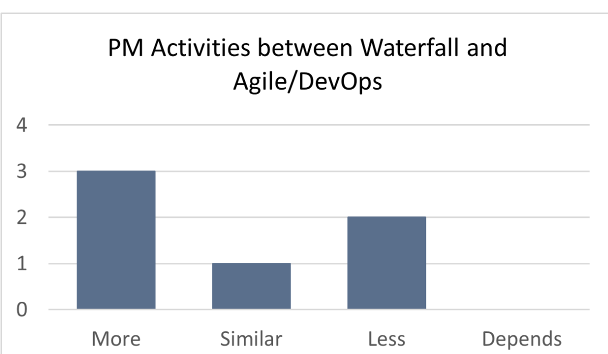


Figure 11. Quantitative Summary of Survey Question 4 Responses

Question: Program Management (PM) is significantly less for Agile/DevOps compared to Waterfall programs. Are you noticing the same behavior? What is causing that (example: reduced deliverables, management activities being moved into development teams)?

This question received mixed answers across the organizations. While the data suggests that PM costs are lower for Agile/DevOps programs compared to Waterfall, the industry partners had different experiences. Three organizations noted that the development team took over some of the PM responsibilities and activities, which leads to a reduction in the PM costs. One organization further noted that the reduction is caused by the developers directly interacting with the Government side of the program, versus going through the PM. Yet, the first two organizations state that the PM activities may have actually increased for Agile/DevOps programs in order to change existing processes and engage the stakeholders regularly.

Question 5: Reduction in Integration and Test?

Question: Integration & Test (IT) is significantly less for Agile/DevOps compared to Waterfall programs. Are you noticing the same behavior? What is causing that (example: integration and testing efforts being captured within development efforts, as they moved into Scrum/development teams)?

Generally, all industry partners are seeing a reduction in IT costs because the activities are either being bucketed with development or because of savings from automated and continuous testing.

Question 6: Improvements in analytical and/or programming skills?

Question: On 2 different datasets, Causal Inference algorithms found a causal link between analyst and programmer capability. From my previous experiences, I found that teams that had good analytical skills also had

Industry Partner	Summarized Answer
Organization 1	Improvements in peer review and test case development. However, not sure analytical/coding skills improved because of Agile or including SE personnel with the software development teams.
Organization 2	Noticed cross-training between the SE and development personnel, and improvements in the knowledge base.
Organization 3	The Agile methodology provides opportunities for developers to demonstrate their skills more compared to Waterfall.
Organization 4	Noticed an increased in productivity with including a SE with the development team.
Organization 5	Noticed an increase in productivity because SE and IT personnel being part of the Scrum team allows issues to be troubleshooted faster.

Table 12. Survey Question 6 Response Summaries

the tendency to be better programmers. Have you noticed if the analytical and/or programming skills of the developers improved with SE and PM FTEs being involved in the sprints/iterations?

Causal Inference algorithms attempt to discover causal relationships from observational data. I applied these algorithms on 2 software development datasets, and the algorithms returned a link between analyst and programmer capabilities in both datasets (though, I did not emphasize or report this result in the studies, as the focus was on causal relationships with effort and schedule) (Hira, Boehm, Stoddard, & Konrad, Preliminary Causal Discovery Results with Software Effort Estimation Data, 2018) (Hira, Boehm, Stoddard, & Konrad, Further Causal Search Analyses With UCC's Effort Estimation Data, 2018) (Alstad, Hira, Brown, & Konrad, 2021). In general, industry agrees that including a system engineer with the Scrum/development teams improves productivity, and that the Agile methodology allows developers to demonstrate and improve their analytical and programming skills.

Question 7: Requirements, Architectures, and Designs Improving?

Question: Has including SEITPM FTEs in the Scrum/development teams led to improved requirements gathering and accuracy, architectures, and designs?

Industry Partner	Summarized Answer
Organization 1	Improvements in requirements gathering, architectures, and designs do not come free with Agile/DevOps. Need a higher-level architecture team.
Organization 2	Really see improvements when stakeholders participate in planning meetings. They are able to clarify and see the requirements.
Organization 3	Noticed less rework, which implies better accuracy. Architecture can depend on external systems and other dependencies, but easier to incorporate changes with Agile/DevOps model.
Organization 4	Not sure (don't have sufficient experience to comment on this)
Organization 5	One program did not start to adopt Agile methodologies until a bit later, but the developers found some of the requirements are not as testable as they could and should have been. Therefore, they are having to rewrite them. Another program started with Agile/DevOps methodologies and found the design is better as a result.

Table 13. Survey Question 7 Response Summaries

One of the Agile Principles states that "the best architectures, requirements, and designs emerge from self-organizing teams" (Beedle, et al., 2001). From the responses received in Table 13, industry generally notices improvements in requirements gathering, designs/architectures, and rework as a result of adopting Agile/DevOps methodologies. However, as the first organization stated, this does "not come free." The improvements depend on having a good architecture team, the team engaging with the stakeholders, and working together to write the requirements.

Question 8: Change in Quality of Products?

Question: Since requirements are gathered and the design/architecture is built incrementally, have you noticed positive or negative changes in the quality of products?

While 2 organizations have experienced both positive and negative changes to product quality, 3 organizations have noticed improvements in product quality as a result of adopting Agile/ DevOps methodologies. While product quality can improve, teams must ensure to not lose focus of the bigger picture and not think of their development environment as a playground.

Question 9: Trouble with Meeting Level of Service Requirements?

Question: Since requirements are gathered and the design/architecture is built incrementally, have you noticed any trouble with meeting level of service requirements later in the development lifecycle, compared to when using the Waterfall lifecycle mode?

"Level of service" requirements refer to requirements that affect the usage of the software systems, such as meeting availability, reliability, scalability, etc. needs. One concern with the design/architecture being built incrementally is whether the architecture/design can and will scale to the needs of the users, especially if these requirements are pushed towards the end of the lifecycle.

While 2 organizations could not comment on this question, the remaining 3 noticed that there is no issue in meeting level of service requirements as long as the discussions, implementing, and testing of these requirements are being done early.

Question 10: Change in Maintainability?

Question: Since requirements are gathered and the design/architecture is built incrementally, has the maintainability of the product improved/ decreased for Agile programs compared to Waterfall?

For this question, maintainability refers to how easily existing software can be modified and maintained. Specific metrics were not required for this question, but just the teams' intuition on how easily they were able to make changes to their existing code.

From the responses, it seems the maintainability of software depends on the system itself and decisions made by the team. This question received varied responses across the participants.

Industry Partner	**Summarized Answer**
Organization 1	Stable, upfront requirements needed for less rework. But Agile can lead to rework.
Organization 2	Fewer defects, because seeing and fixing earlier.
Organization 3	Decrease in rework and less defects. Comes down to overall design, complexity of programs, and maturity of teams.
Organization 4	No answer
Organization 5	Same, but earlier in the lifecycle.

Table 14. Survey Question 11 Response Summaries

Question 11: Change in Rework and Defects?

Question: Since requirements are gathered and the design/architecture is built incrementally, have you noticed reduction/increase in rework, scrapped code, and defects?

Agile/DevOps teams are noticing fewer defects at the end of the lifecycle, because defects are being noticed and fixed earlier. Only 1 organization provided insight on rework, which seems to depend on the stability of requirements.

Conclusion

From the survey responses received from and follow-up discussions with industry, the phenomena and insights that are mostly common across the 5 organizations are:

- Software development/Scrum teams are cross-functional: SE and IT full-time equivalents (FTEs) are generally included.
- SEITPM activities/effort/costs are level-loaded across the lifecycle.
- IT costs are lower due to the activities being bucketed with development, and due to savings from automated and continuous testing.
- Including SE FTEs with development/Scrum teams leads to higher productivity.
- Organizations have noticed an improvement in requirements gathering, architectures and designs from adopting Agile/DevOps methodologies.
- There is an improvement in the product quality, though a couple organizations mentioned that they have also had scenarios where there was a negative impact.
- The organizations have not faced challenges in meeting level of service requirements as long as the discussions, implementation, and testing of these requirements are being done early.
- There are fewer defects at the end of the lifecycle because they are found and fixed earlier. The amount of rework required depends on the stability of requirements, however.

However, industry, as a whole, did not have unified or strong insights for the remaining 3 questions in the survey.

The goal of the survey questions was to ask industry if they noticed the reduced SEITPM costs in Agile/DevOps environments and whether that led to positive or negative effects in the final products. In general, organizations and software development teams noticed reductions in Integration and Test (IT) costs most significantly. Though the data suggests Program management (PM) costs are also lower for Agile/DevOps programs compared to Waterfall programs, industry did not necessarily notice a decrease in the PM activities. The organizations also noticed mostly positive effects in product quality, defects, and meeting level of service requirements. While improvements were not necessarily noticed for rework and maintainability, they also did not necessarily worsen compared to Waterfall programs.

Threats to Validity

This research study is based specifically on Ground software systems from the Space Systems Command (SSC) and Air Force. The programs range from new development to modifications to existing systems and vary in terms of functionality provided and sizes. Given the nature of the data used in this study, there are 2 threats to validity:

1. Since the data and survey participants come from Ground systems, the findings in this study might not apply to other application domains (particularly the SEITPM costs estimating regression (Equation 2)). As mentioned in the Future Work section (below), a good future step would be to analyze data across different application domains/types to evaluate how generalizable the findings are.

2. The Agile/DevOps programs in the datasets used in this study have a considerably small total cost/size range compared to the Waterfall programs. Therefore, the results in this paper might not hold for larger, more complex programs using the Agile/DevOps methodologies. Also mentioned in the Future Work section (below) is the suggestion to update this study when larger Agile/DevOps data is collected to observe whether the SEITPM costs are still lower than for the Waterfall programs.

Comprehensive Conclusions

This research study consists of 2 parts:

1. Analyze the SEITPM costs between Agile/ DevOps and Waterfall programs

2. Survey industry to get their insights on the SEITPM cost differences between Agile/ DevOps and Waterfall programs.

The first part of the study showed that the SEITPM costs are about 30% less for Agile/ DevOps programs compared to Waterfall programs and that this difference is statistically significant. By looking at the individual activities separately, the reduction in PM and IT costs Agile/DevOps and Waterfall programs are statistically significant, while the reduction in SE costs is not.

Reduced SEITPM costs can imply insufficient systems engineering and planning activities, which can lead to the program's inability to scale to requirements, increased defects, reduced maintainability of the code, and overall decline in products' quality. The second part of the research study, surveying and having discussions with industry, was designed to understand whether the development teams are noticing a decline in product quality as a side-effect to adopting Agile and/or DevOps methodologies.

Discussions with industry concluded that the software development teams usually did not notice a major reduction in SEITPM costs and activities for Agile/DevOps programs – particularly for SE and PM. One thing to note here is that the industry partners did not study their own data prior to these discussions and were asked to answer based on their intuition. This suggests that there is not an active attempt to reduce SEITPM activities because maintaining product quality is essential. However, they did note that the responsibilities, activities, and cost reporting between software development and SEITPM activities had blurred and overlapped more than on Waterfall programs. In general, the industry noticed either an improvement or similarity in the product's quality, number of defects, rework, and maintainability compared to Waterfall programs.

In answer to the question posed in the title of this paper (are Agile/DevOps programs doing enough systems engineering?), this research study found that software development teams are able to and have been doing enough engineering to produce high quality products while utilizing Agile/ DevOps methodologies and reducing costs.

Future Work

There are several future steps that could enhance this analysis further:

1. Perform a similar analysis on a dataset that contains data points across the various application domains to evaluate whether the findings in this study are generalizable.

2. Reach out to more industry teams, not just SSC's industry partners, to get their responses on the survey questions. With more responses, we may be able to understand if there are patterns that are more common than others as well as all the unique ways Agile/DevOps teams are formed.

3. Collect annual SEITPM costs across multiple Waterfall and Agile/DevOps programs to be able to generalize how the SEITPM activities' levels behave throughout the lifecycle and how they differ between the 2 groups.

4. Update this analysis when more data on Agile/DevOps programs is collected, especially on larger programs. Since the Agile/DevOps programs are significantly smaller than many of the Waterfall programs in the data used, it is unclear if the behavior identified (that SEITPM costs are significantly lower for Agile/DevOps programs compared to Waterfall programs) will continue as the Agile/DevOps programs grow in size and difficulty.

Acknowledgements:

The author thanks the Air Force Cost Analysis Agency (AFCAA) for sharing the data used in this research study, and Raj Palejwala, Natasha Edwards, Adriana Contreras, and Ernest Rangel of the Space Systems Command (SSC) for supporting and reviewing this study. The author also thanks Matt Murdough, Miguel Aceves, and Ben Kwok of Tecolote Research Inc. for reviewing, providing suggestions, and guiding this study from initiation to submission. This study was funded by the Space Systems Commands (SSC) under contract FA8802-19-F-0005.

References:

Alstad, J., Hira, A., Brown, A. W., & Konrad, M. (2021). Investigating Causal Effects of Software and Systems Engineering Effort. *International Cost Estimating and Analysis Association Professional Development and Training Workshop.*

Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering and Technology, 2*(5).

Beedle, M., van Benekum, A., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., . . . Thomas, D. (2001). *Principles behind the Agile Manifesto*. (Agile Manifesto) Retrieved November 5, 2021, from https://agilemanifesto.org/principles.html

Ben-Zahia, M. A., & Jaluta, I. (2014). Criteria for selecting software development models. *2014 Global Summit on Computer \& Information Technology (GSCIT)* (pp. 1-6). IEEE.

Davis, G. (2000). Managing the test process [software testing]. *Proceedings International Conference on Software Methods and Tools. SMT 2000* (pp. 119-126). IEEE.

Hira, A., Boehm, B., Stoddard, R., & Konrad, M. (2018). Further Causal Search Analyses With UCC's Effort Estimation Data. *Acquisition Research Program.*

Hira, A., Boehm, B., Stoddard, R., & Konrad, M. (2018). Preliminary Causal Discovery Results with Software Effort Estimation Data. *Proceedings of the 11th Innovations in Software Engineering Conference.*

Hu, S.-P. (2001). Minimum-Unbiased-Percentage Error (MUPE) Method in CER Development. *Third Joint Annual ISPA/SCEA International Conference.*

Manring, J. (2016). Maturing the Economic Aspects of Agile Development in the Federal Government. *International Cost Estimating and Analysis Association Professional Development and Training Workshop.*

Markman, M. R., Ritschel, J. D., & White, E. D. (2021). USE OF FACTORS IN DEVELOPMENT ESTIMATES: IMPROVING THE COST ANALYST TOOLKIT. *Defense Acquisition Research Journal: A Publication of the Defense Acquisition University, 28*(1).

Russo, D. (2021). The Agile Success Model: A Mixed-methods Study of a Large-scale Agile Transformation. *ACM Transactions on Software Engineering and Methodology (TOSEM), 30*(4), 1-46.

Seaver, D. (2018). Agile to DevOPS and its Impact on Estimation and Measurement. *Joint IT and Software Cost Forum.*

Sinha, A., & Das, P. (2021). Agile Methodology Vs. Traditional Waterfall SDLC: A case study on Quality Assurance process in Software Industry. *2021 5th International Conference on Electronics, Materials Engineering \& Nano-Technology (IEMENTech)* (pp. 1-4). IEEE.

Tracy, S. P., & White, E. D. (2011). Estimating the final cost of a DoD acquisition contract. *Journal of Public Procurement*.

Anandi Hira *is currently a Data Scientist/Researcher at the Carnegie Mellon University Software Engineering Institute (CMU SEI). Previously, Anandi performed several Agile and software cost estimation research projects as a cost analyst at Tecolote Research Inc. She received her PhD in software cost estimation under Dr. Barry Boehm at University of Southern California (USC), where she collected data and calibrated the COCOMO® II model to include functional size metrics. Her research interests include software metrics and its application to project management, software cost estimation, and software process improvement.*

Projecting Future Costs with Improvement Curves: Perils and Pitfalls

Brent M. Johnstone

Abstract: Improvement curves are one of the most common projection tools used by cost estimators. Their use is surrounded however by perils and pitfalls. Common errors include: the fallacy of "straight edge and graph paper" projection, the dangers of recovery slopes, failure to understand how development and production environments differ, and the dangers of using learning curve slopes to measure production line efficiency. This paper examines these potential pitfalls and proposes ways to avoid them.

Introduction

Improvement curves are one of the most common tools that cost estimators use to project future costs. Unlike a ladder or power tool bought at the hardware store, improvement curves do not come with warning labels. Perhaps they should: the consequences of misusing them can be quite significant. The stakes of a bad cost estimate can be high – millions or even billions of dollars in funding or profits may depend on decisions estimators make. Consider this paper in some sense a warning label: it identifies the perils and pitfalls of improvement curves and looks at common errors in projecting future costs based on the author's experience in the military aircraft industry.

This paper will examine five potential perils:

- the peril of straight-line projection
- failure to account for the impacts of development versus production
- the dangers of recovery slopes
- carelessness about designating the first unit
- dangers of using learning curve slopes to measure production line efficiency

Peril: The Straight-Line Projection

A common method of projection using the learning curve is to regress historical data, calculate the curve slope, then assume that same slope to project the cost of future work. “You are on an 83% learning curve,” the analyst announces as if he is stating an inviolable law of nature. “You should be on the same slope for future lots.” Proof that this slope is valid for future projection is typically buttressed by a statement of the regression line’s R^2 – the higher the R^2 the “better” the model and the more certain the future projection. This can be called the “straight edge and graph paper” school of estimating – projecting the future is no more difficult than drawing a best fit line on log-log paper and projecting that line through the number of units being estimated.

What could be wrong with this? Empirical studies have demonstrated that this is in fact is not a reliable method to project future costs. Dutton (1984) cautioned:

> In general, the empirical findings caution against simplistic uses of either industry experience curves or a firm’s own progress curves. Predicting future progress rates from past historical patterns has proved unreliable.

Similarly, Fox, et al. (2008) cited:

> Even with both an excellent fit to historical data (as measured by metrics like R^2), and meeting almost all of the theoretical requirements of cost improvement, there is no

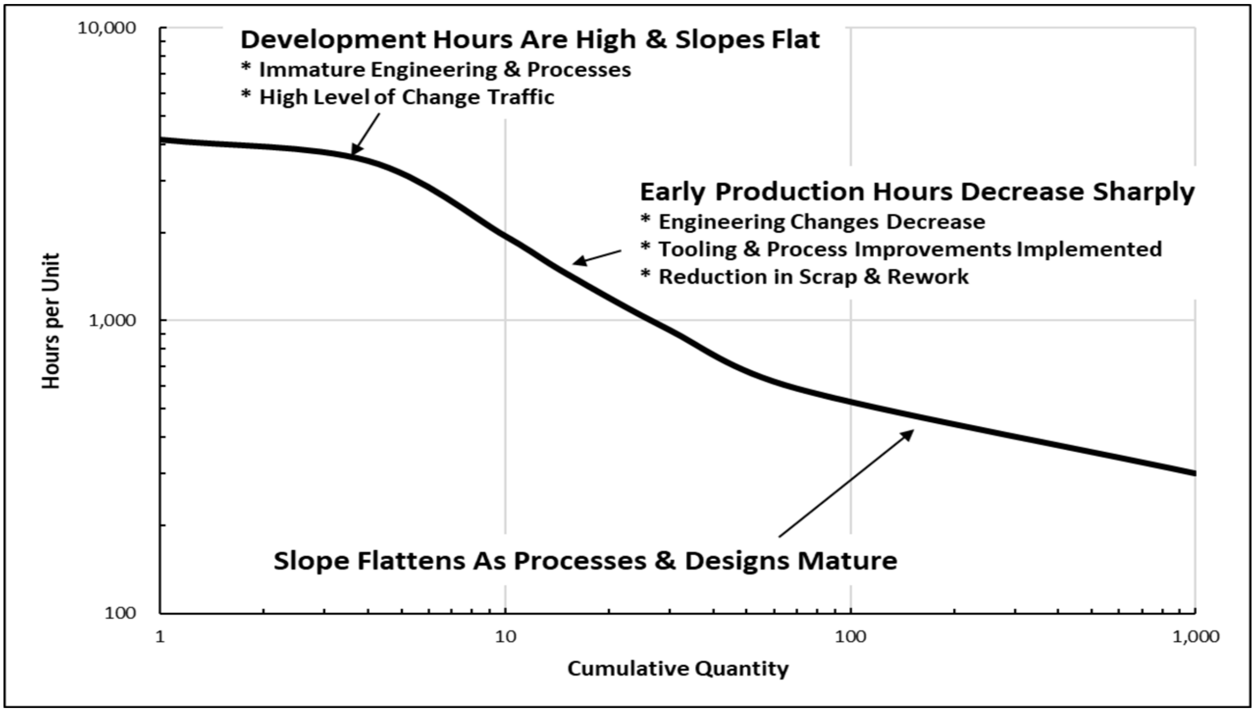


Figure 1. Profile of the S-Curve (Notional)

> guarantee of accurate prediction of future costs….[E]ven projections based on producing an almost identical product over all lots, in a single facility, with large lot sizes, and no production break or design changes, do not necessarily yield reliable forecasts of labor hours.

Continuing, Fox, et al. writes:

> Out-of-sample forecasting using early lots to predict later lots has shown that, even under optimal conditions, labor improvement curve analyses have error rates of about +/- 25 percent.

The primary reason for this failure is that the learning curve is frequently not a straight line in log-log space over the product life cycle. The initial learning curve studies (Wright, 1936; Crawford, 1944) understood improvement curves as straight-line logarithmic functions. Within a few years, however, observers began to see improvement curves not as straight lines in a log-log space, but curvilinear functions that exhibited an "S" shape based on product and process maturity (Carr, 1946; Stanford Research Institute, 1949; Asher, 1956; Cochrane, 1960; Cochrane, 1968).

The S-shaped improvement curve as commonly drawn is composed of three stages, captured graphically in Figure 1 (Carr, 1949; Cochrane, 1960; Cochrane, 1968).

The first stage, typically in the product development phase, shows high hours per unit and relatively flat improvement curve slopes. The limited degree of improvement is caused by an evolving engineering design and immature manufacturing processes. Part shortages disrupt the continuity of production. Scrap and rework is high, and there are typically a high number of engineering changes.

In the second stage, typically during early production, the hours per unit decrease sharply along a relatively steep improvement curve. The production rate increases significantly from the relatively low delivery rates of the development phase. Engineering changes decrease sharply, while improvements in tooling and manufacturing processes are implemented. Manufacturing scrap and rework also decreases at a faster rate. Shortages decrease as the supply chain begins efficiently feeding the production line.

In the third stage, production rates continue to increase to their maximum build rate. Manufacturing processes, tooling and engineering designs mature. Consequently, the pace of production improvements slow and the learning curve slope flattens in response. (Boone, 2021)

The easiest way to understand the changing curve slope over time is to understand the definition of the learning curve itself. A Northrop publication from the 1960's defines the learning curve as "the rate at which management identifies and solves problems in relation to design, methods, shortage of parts, inspection and shop education." (Jones, 2001) Logically, the rate at which problems are solved will change over time – the "low hanging fruit" with the fastest payoffs will be picked first, leaving the more intractable and difficult problems to be solved later, or maybe not at all.

What is the significance for our estimator? If he does not consider where he is in the product life cycle but blindly continues the historical slope, he may significantly overstate or understate future hours. (Reference Figure 2.) If his history is from the initial development stage, he may miss the steepening which typically occurs in the early production stage and overstate his estimate. If his history is from the early production stage, he may miss the flattening that occurs as product designs and manufacturing processes mature and understate his estimate. This does not mean that the analyst should never project a historical slope forward. Suppose the program has reached full production and its engineering and manufacturing processes are mature. In such a case it might be appropriate to project the next production lot by continuing the historical slope. But these decisions cannot be made carelessly

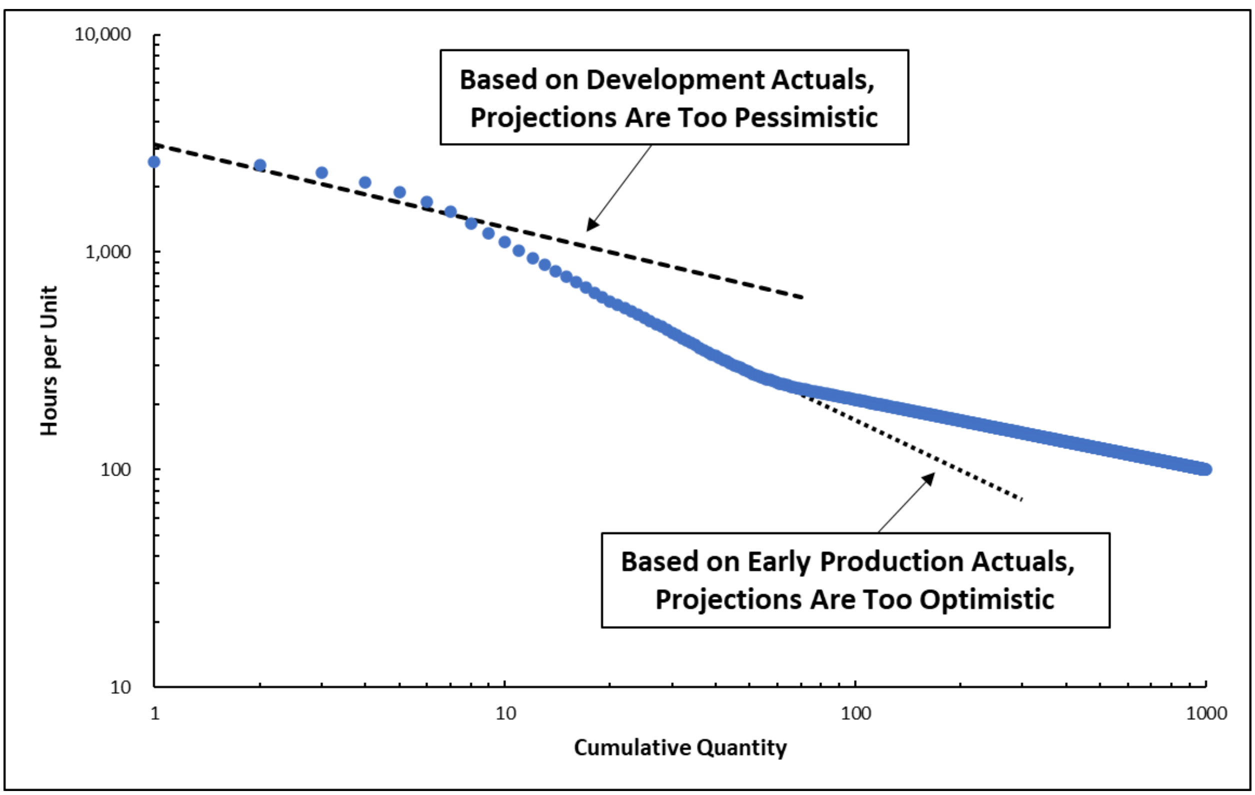


Figure 2. S-Curve & Impact on Projections (Notional)

without understanding where historical experience falls along the product life cycle.

But what about those sterling best fit statistics our analyst quoted earlier? True, many writers on learning curves recommend using best fit statistics as a criterion for choosing a particular learning curve slope and theory -- but as *a* criterion, not the *sole* criterion. Nussbaum and Mislick (2015) introduce numerous factors which should be considered in determining learning curve slopes including the nature and quality of production tooling; supplier competence and experience; expected number of design changes; length of part lead times; similarity of the product to other systems; and historical experience across the product lifecycle. These factors can and do change over the course of a production program.

Using R^2 blindly to justify continuing a straight-line projection – on the basis that past is prologue – recalls the metaphor of driving a car by only looking through the rear-view mirror. Schumeli (2010) distinguishes sharply between explanatory models and predictive power. R^2 is a statistic which explains the historical association between the variables of a model. It can make no justifiable claim about the future. As Schumeli notes, models which do a good job of explaining *observed* behavior may do a poor job of predicting *future* behavior.

Continuing on this theme, Schumeli writes:

> Researchers report R^2-type values and statistical significance of overall F-type statistics to indicate the level of explanatory power. …A common misconception in various scientific fields is that predictive power can be inferred from explanatory power. However, the two are different and should be assessed separately. … Measures such as R^2 and F would indicate the level of association, but not causation. …In general, measures computed from the data to which the model was fitted tend to be overoptimistic in terms of predictive accuracy: "Testing the procedure on the data that gave it birth is almost certain to overestimate performance." (Mosteller and Tukey, 1977)

Regardless of the historical R^2, if a regression model ignores product and manufacturing maturity and their associated cost impacts, it will not do a good job of predicting the future.

Solution: Using Multiple-Leg Curves Prevents "Straight Edge" Fallacy

Without actual cost history, analogous program data combined with analysis of the programmatic factors previously referenced by Nussbaum and Mislick can be used to derive the projected learning curve slopes and breakpoints to project a S-shaped improvement curve. My earlier paper on improvement curves (Johnstone, 2015) suggests a methodology for early production when there are limited actual cost history. In this instance, let us assume there is sufficient historical data on the program in question, and a change in slopes can be inferred from a visual inspection of the data.

There are several learning curve models which allow an S-shaped improvement curve to be derived (Miller, 1971; Jones, 2001). This paper suggests a discontinuous regression model which can be easily built from historical data.

We start from our familiar improvement curve model:

$$y = \alpha_1 x^{\beta_1} \tag{1}$$

Where:

y = Manufacturing hours per unit

x = Cumulative units built to date

α_1 = Y-intercept, equal to theoretical first unit (TFU) hours

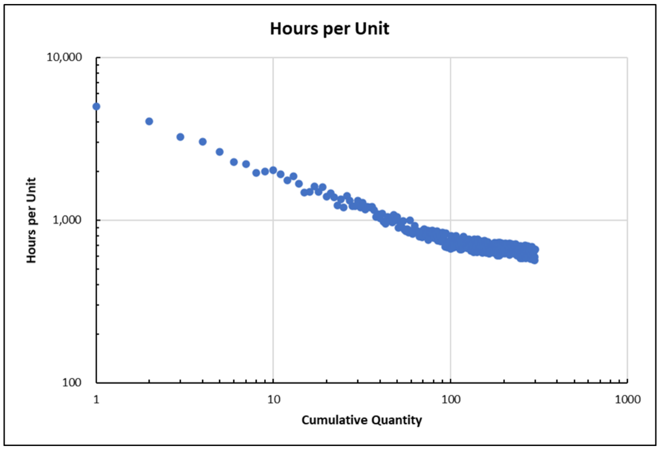


Figure 3. Notional Data Set

β_1 = Rate of learning, such that 2^{β} equals learning curve slope

After hours per unit and cumulative quantities are converted to natural logarithms, this yields the following linear form:

$$\ln y = \ln \alpha_1 + \beta_1 \ln x \qquad (2)$$

Kennedy (1992) outlines a method for using dummy variables to capture a change in the intercept and slope coefficients between two periods. To create a two-leg segmented learning curve, we introduce breakpoint unit *T*. Based on our *a priori* selection for *T*, data is separated into pre-break period 1 ($x < T$) and post-break period 2 ($x \geq T$). In addition, dummy variable *D* is created such that *D* is zero for period 1, and one for period 2. Product dummy variable *Dx* is also created such that *Dx* takes the value *x* in period 2 but is 0 otherwise. This creates the regression equation:

$$\ln y = \ln(\alpha_1 + \alpha_2) + (\beta_1 + \beta_2) \ln Dx \qquad (3)$$

Equation (3) represents two separate cases. Where $x < T$, variables *D* and *Dx* are 0 and equation (3) reduces back to our standard improvement curve equation (2). But where $x \geq T$ and *D* takes the value of one, different intercept and slope values are introduced such that:

$$\ln y = \ln \alpha_1 + \ln \alpha_2 D + \beta_1 \ln x + \beta_2 \ln Dx \qquad (4)$$

Where:

y = Manufacturing hours per unit (HPU)

α_1 = Y-intercept for leg #1, equal to theoretical first unit hours for leg #1

α_2 = Intercept adjustment for leg #2, such that $\alpha_1 + \alpha_2$ equals the Y-intercept for leg #2

β_1 = Rate of learning for leg #1, such that 2^{β} equals learning curve slope #1

β_2 = Rate of learning for leg #2, such that $2^{(\beta 1 - \beta 2)}$ equals learning curve for leg #2

To demonstrate how such a curve can be built, a notional data set was constructed as follows. Based on a visual inspection of Figure 3, unit 101 was chosen as the breakpoint *T*. To illustrate further, a table of selected units (Table 1) is displayed to show *D*, *x* and *Dx.* Finally, a sample output from Microsoft Excel (Figure 4) is shown after selecting the natural logarithm of hours per unit as the dependent variable *y* and regressing *D*, *x* and *Dx* as independent variables.

We may interpret the results as follows: For units 1-100, hours per unit are calculated using a theoretical first unit (TFU) of 5,192 hours and a slope of 74.3%. For units 101-300, hours per unit are calculated using a TFU of 1,497 hours (calculated as $e^{(8.555 - 1.244)}$ or $\alpha_1+\alpha_2$) and a slope of 89.9% (calculated as $2^{(-0.428 + 0.274)}$ or $2^{(\beta 1 + \beta 2)}$). This equation also has a high R^2 of 0.97 – which significantly fits the historical data better than an equivalent single slope learning curve (R^2 = 0.925).

It can be argued that the R^2 of the best fit of a discontinuous line will always show some improvement, however miniscule, over the best fit of a single line. To test the statistical significance of the parameter values for period 1 (pre-break) and period 2 (post-break), a Chow test can be performed in the format suggested by Kennedy (2002). By comparing the sum of squared errors (SSE) of the regressions for a

Table 1. Notional Data Table (Partial).

					Dependent Variable	Independent Variables		
Unit	HPU	LN(Unit)	T	LN(T)	LN(HPU)	LN(x)	D	LN(Dx)
1	5,020	-	101	4.62	8.52	-	-	-
2	4,065	0.69	101	4.62	8.31	0.69	-	-
3	3,248	1.10	101	4.62	8.09	1.10	-	-
4	3,038	1.39	101	4.62	8.02	1.39	-	-
5	2,628	1.61	101	4.62	7.87	1.61	-	-
6	2,272	1.79	101	4.62	7.73	1.79	-	-
7	2,216	1.95	101	4.62	7.70	1.95	-	-
8	1,949	2.08	101	4.62	7.58	2.08	-	-
9	2,001	2.20	101	4.62	7.60	2.20	-	-
10	2,030	2.30	101	4.62	7.62	2.30	-	-
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
99	682	4.60	101	4.62	6.53	4.60	-	-
100	668	4.61	101	4.62	6.50	4.61	-	-
101	798	4.62	101	4.62	6.68	4.62	1	4.62
102	677	4.62	101	4.62	6.52	4.62	1	4.62
103	724	4.63	101	4.62	6.59	4.63	1	4.63
104	692	4.64	101	4.62	6.54	4.64	1	4.64
105	680	4.65	101	4.62	6.52	4.65	1	4.65
106	746	4.66	101	4.62	6.61	4.66	1	4.66
107	799	4.67	101	4.62	6.68	4.67	1	4.67
108	724	4.68	101	4.62	6.59	4.68	1	4.68
109	763	4.69	101	4.62	6.64	4.69	1	4.69
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

single slope as opposed to a multiple leg slope, we can create an F-statistic of the form:

$$\frac{[SSE(combined)-SSE(separated)]/K}{SSE\ (separated)\ /\ (T_1+T_2-2K)} \quad (5)$$

The "combined" dataset represents the residuals for the single leg curve, while the "separated" dataset represents the residuals for the two-leg curve with its dummy variables representing pre- and post-break datasets, where *K* represents the number of parameters (including the intercept) of the combined dataset, T_1 the number of observations in period 1, and T_2 is the number of observations in period 2.

The resulting test-statistic derived from equation (5) can be evaluated against a F-table at the desired level of error with *K* and T_1+T_2-2K degrees of freedoms. Our null hypothesis – that $\alpha_1 - \alpha_2 = 0$ and $\beta_1 - \beta_2 = 0$ – would conclude there is no significant structural break in the hours data to justify a two-leg curve. The alternate hypothesis – that $\alpha_1 - \alpha_2 \neq 0$ and $\beta_1 - \beta_2 \neq 0$ – would conclude just the opposite: there is a significant structural break in the data beginning at unit *T*. In our notional example (calculations not shown here but available upon request), we can reject the null hypothesis with a 99.9% confidence, concluding that our data does indeed show a break in the learning curve slope.

As noted above, a high R^2 – even one buttressed by a sufficiently high F-statistic for the Chow test -- does not guarantee the accuracy of the forecasts made from this equation. But this two-leg model is more in line with the theoretical expectations set by the S-curve as well as historical experience, and therefore more likely to give us a better projection of future costs.

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.985
R Square	0.971
Adjusted R Square	0.970
Standard Error	0.058
Observations	300

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	3	32.352	10.784	3,249.084	0.000
Residual	296	0.982	0.003		
Total	299	33.335			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Natural log - Intercept (ln α1)	8.555	0.023	365.375	-	8.509	8.601	8.509	8.601
Natural log - Beta-1 (ln β1)	(0.428)	0.006	(68.580)	0.000	(0.440)	(0.416)	(0.440)	(0.416)
Natural log - Alpha-2 (ln α2)	(1.244)	0.074	(16.877)	0.000	(1.389)	(1.099)	(1.389)	(1.099)
Natural log - Beta-2 (ln β2)	0.274	0.015	18.706	0.000	0.245	0.303	0.245	0.303

Results:

5,192	TFU for Leg #1
74.3%	Slope for Leg #1
1,497	TFU for Leg # 2
89.9%	Slope for Leg #2

Figure 4. Microsoft Excel Output

Peril: Development Slopes – Ignorance Is Not Bliss

One of the conclusions of the S-curve theory is that improvement slopes in a development phase of a program will be relatively flat. This is because so many programmatic issues conspire together to prevent rapid improvement in costs. These include very high number of engineering changes, late parts usually due to late engineering release, tooling which requires rework, engineering errors, and the realization that manufacturing processes and part flows that work on the drawing board don't necessarily work on the shop floor. It's hard to overstate the chaos of the start-up of a manufacturing line. It's a recurring theme on many programs that parts are installed in one station only to be ripped out and replaced just a few stations further down the line because of engineering changes.

Yet in much of the learning curve literature this tends to be glossed over. In many surveys the data from the development units is either excluded, or data limitations prevent an analysis of development slopes. For example, in RAND's 2001 study of military fighter aircraft (F-14, F-15, F-16, F-18, AV-8B), Engineering and Manufacturing Development (EMD) data is included as a single aircraft lot, not as individual units. This prevents any analysis of a unique EMD slope. RAND's conclusion that the average improvement slope for manufacturing is 80% therefore tells us little about the shape of the improvement curve in the development phase itself (Younossi, 2001). Similar issues plague other industry-wide studies (Resetar, 1991; Hess, 1987; Levinson, 1966). However, insight into individual unit cost data is often only found in company-proprietary datasets. Only at the individual unit level does the slower rate of improvement for the development phase becomes apparent.

So why does this matter? Because decisions which are made about the slope of the initial units can be critical to establishing the eventual production cost.

Let us take a simple example. An estimator establishes the cost of a 300-unit program using an S-curve profile. For the ten-unit development phase, he projects using an 86% slope. When production begins at unit 11, the slope steepens to a 72% slope which is maintained until T-101, at which point it flattens to 82%. When the estimate starts running through the company approval cycle, however, the program manager objects.

For one thing, the program manager doesn't like the idea of a three-leg curve. Shouldn't a learning curve be a single line? Moreover, a relatively flat development slope might appear uncompetitive to the source selection committee. The discussion goes on for several minutes, until the program manager suggests that the program use the same T-1 and T-300 costs as originally proposed but simply draw a single slope in log-log space between those two points. The program manager recognizes that the development phase will be initially understated, but it is only for ten units, after all, and it might put the company in a better competitive position.

Figure 5 illustrates the program manager's solution. Unfortunately, this solution does not just put the development cost estimate at risk. It also significantly understates the cost of the first three production lots.

While it is true that the gap between the two approaches begins to close at Lot 3, the damage has been done. If the analyst's original estimate was right, the first two production lots will overrun by 21% and 15% respectively. This could lead to adverse publicity, and the perception that the program is unable or unwilling to control its costs. It could also lead to a substantially degraded financial position for the company. If the original estimate is wrong – and history says the odds it will be too low are far better than being too high – then the damage to the program and to the company will be even greater.

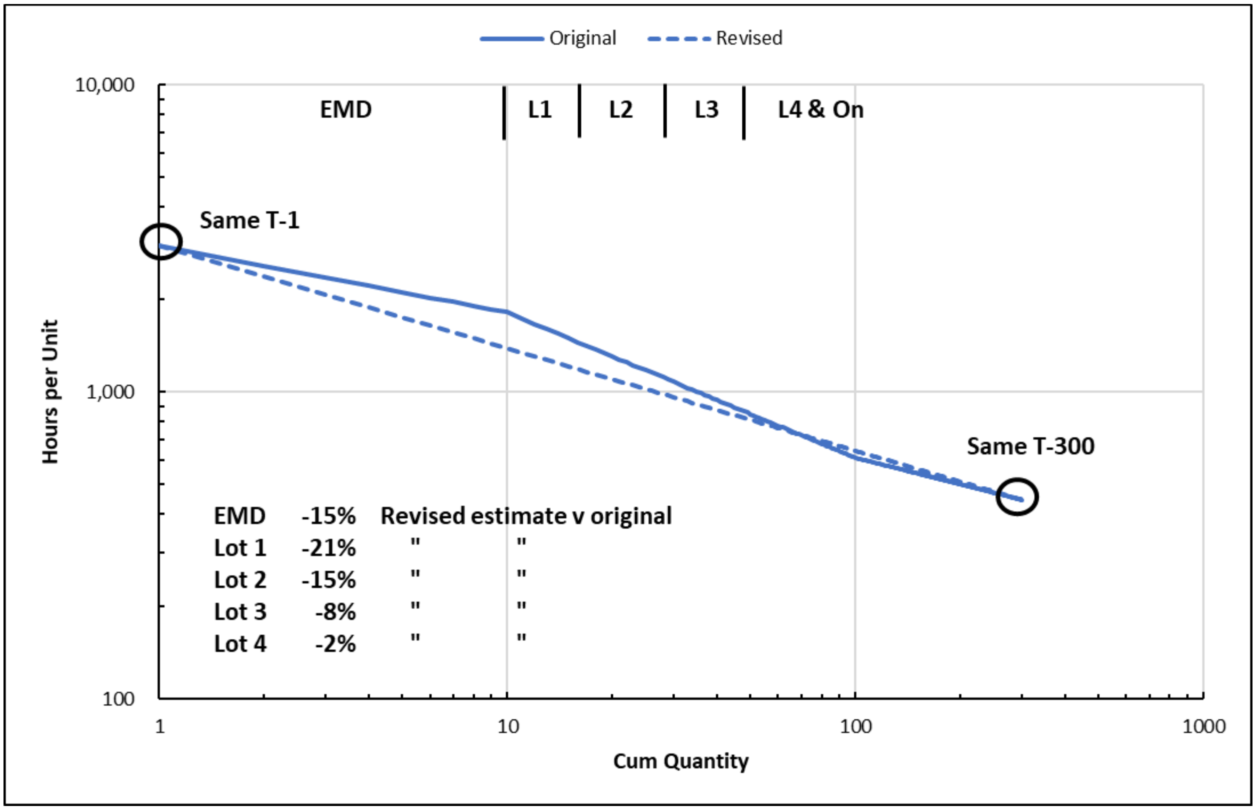


Figure 5. Impact of Flatter Development Slope on Performance (Notional)

Solution: Recognize That Choices Matter

The solution here is relatively simple – be aware that choices about learning curve slopes during the development phase impact the estimate. These choices can be consequential whether the program follows a straight-line logarithmic function or an S-curve pattern.

One word on the assumption that development programs always have relatively flat improvement slopes: It is true that you can sometimes find development programs which have a steep improvement curve. In the author's experience, such an occurrence is typically due to an unusually high first unit cost, which in turn is driven by programmatic issues. Programs that push the manufacturing state of the art by introducing new or radical processes often show high first unit costs as companies struggle to implement these on the shop floor. This poor performance is typically followed by rapid cost improvement as issues are worked through. The Convair B-58 program, built in the 1950's and 1960's, provides an example. Not only was the B-58 the first supersonic bomber, but it introduced the first widespread use of honeycomb bonded structure (Hess, 1987). Issues with the fabrication of the panels and their subsequent installation led to a high first unit cost but a rapid movement down the learning curve for follow-on units (Large, 1974). These examples are the exception, however, and not the rule.

Peril: The "Slippery Slope" – Extraordinary Impacts and Recovery Slopes

One of the most vexing situations for an estimator are those cases where there are sharp increases in unit cost over time -- but the increases are expected to be mitigated over time. These can be divided roughly into two camps: (a) "expected" disruptions, such as major engineering changes,

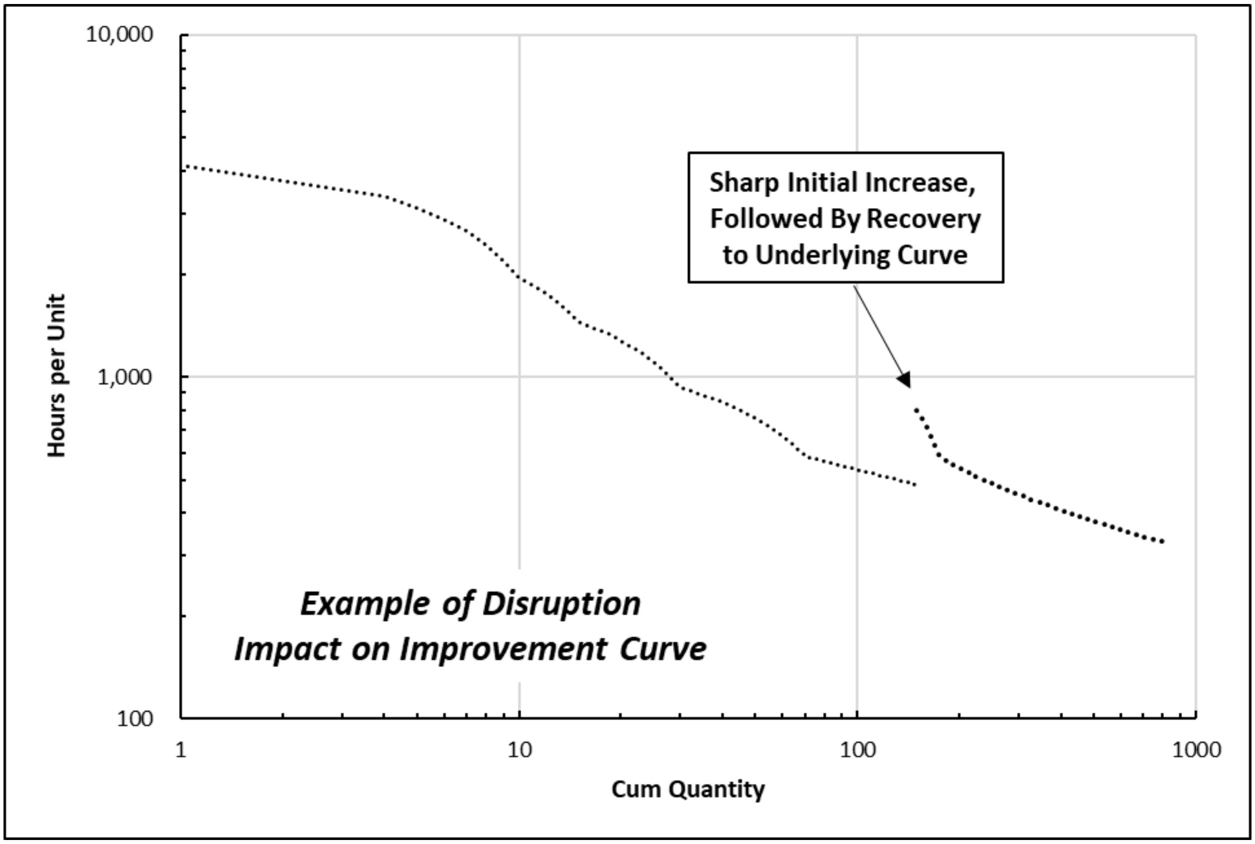


Figure 6. Disruption Example (Notional)

production breaks or work transfers between sites and (b) "unexpected" disruptions caused by unforeseeable circumstances. An example of an unexpected disruption would be a critical load part shortage which creates significant behind schedule and out of station costs. Both types of disruptions appear similarly on graphs of historical costs. Figure 6 shows an example of this kind of behavior, with a sharp initial increase in cost and an eventual asymptotic recovery to the underlying curve.

Of course, *ex ante* we do not have the advantage of how and when this recovery will occur. Herein is our estimating dilemma. How might we deal with this issue?

This is best illustrated by an example. At unit 150 a severe part shortage produces a substantial behind schedule position with workarounds and significant out of station work. This situation (See Figure 7) is expected to end at some point but no one can say with confidence when.

There are two often-taken approaches to this. The first is to simply ignore these units and project the cost as if these impacts had never occurred (See Figure 8). This is often justified by a claim that it represents where the company "should be" performing had the extraordinary impact not occurred. Whether the extraordinary impact is anticipated (e.g., driven by an engineering change or a production break) or unexpected (e.g., driven by part shortages or schedule problems), this procedure is never justified. Assuming away these type of cost increases may seem like a viable approach to the cost estimator. It is never one to the shop floor managers and directors who cannot deal with the world as we wish it was, but as it is. This approach often creates an insurmountable gap between current performance and what the analyst thinks the values "should be."

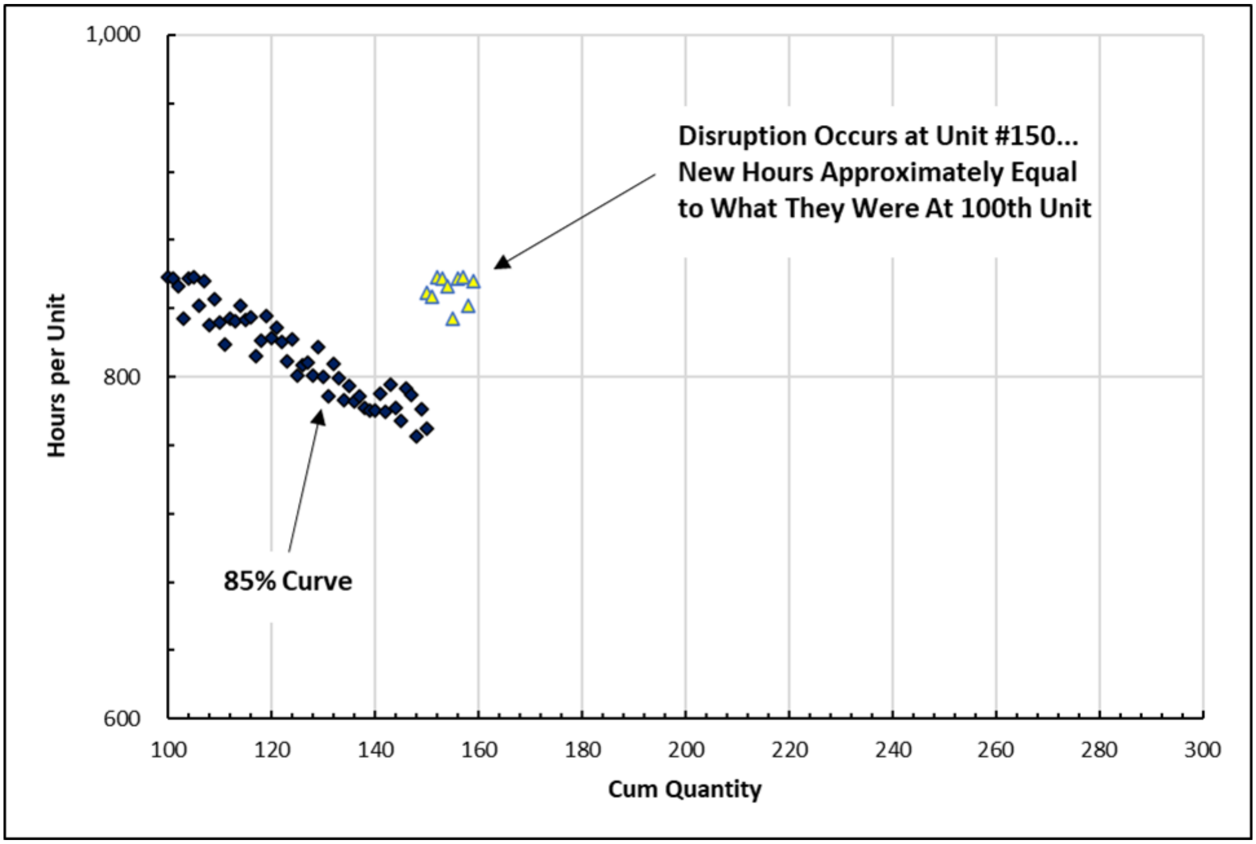


Figure 7. Illustration of Disruption (Notional)

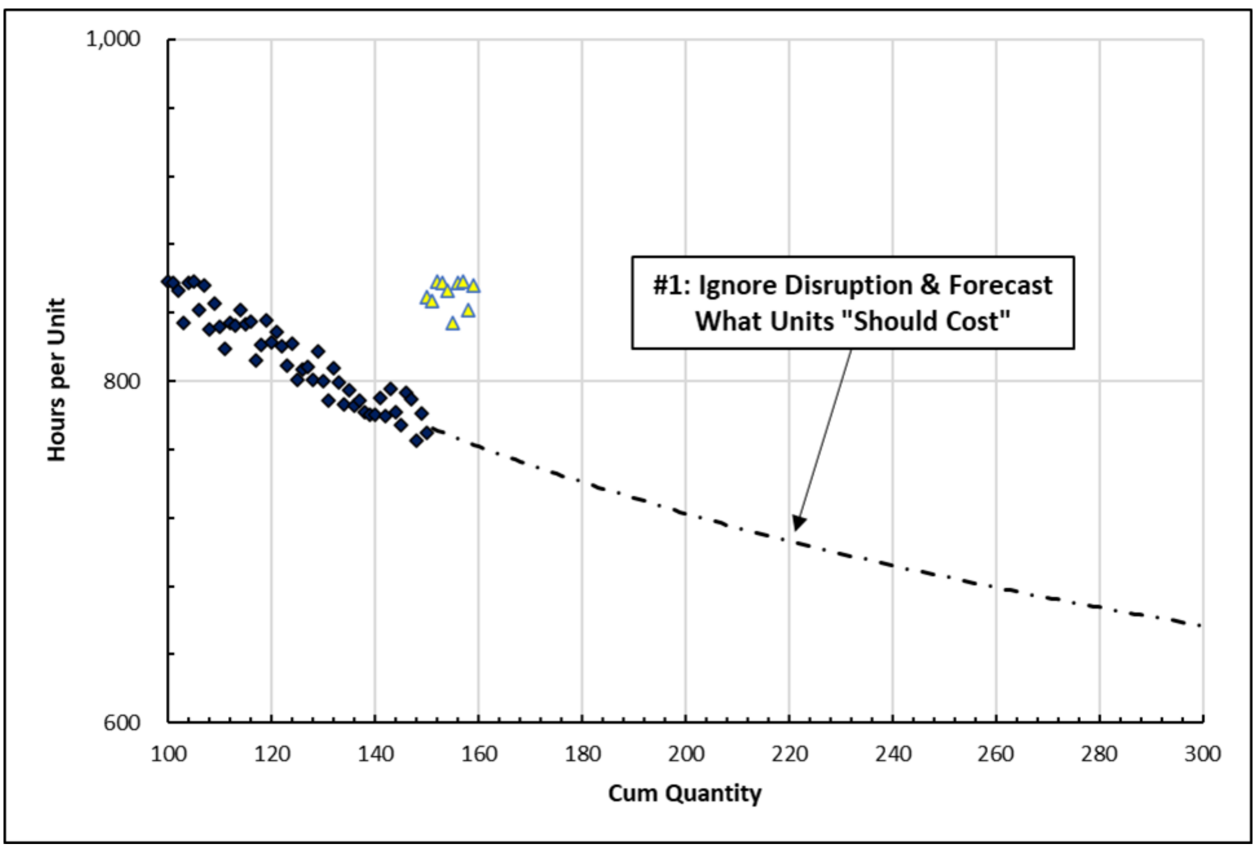


Figure 8. Recovery Curve – Doing Nothing

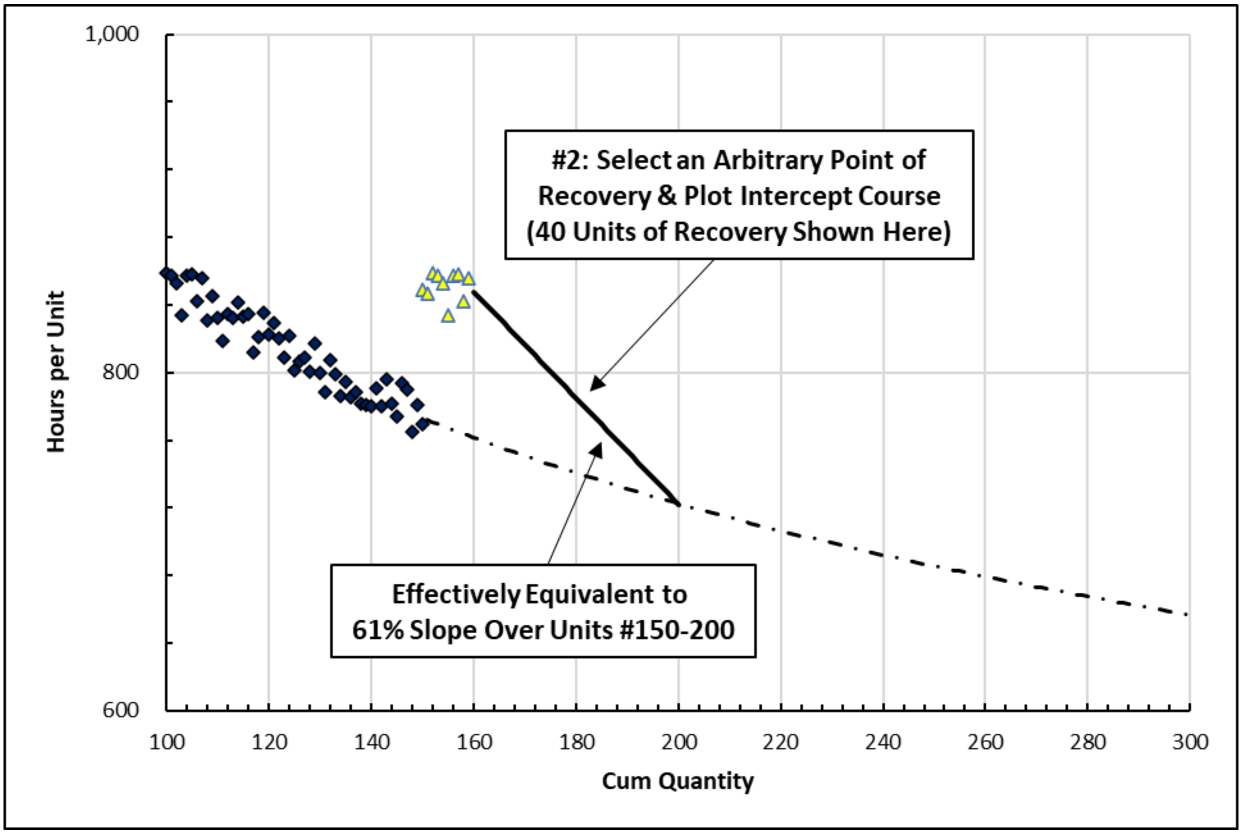


Figure 9. Recovery Curve - Point of Recovery

Unfortunately for the shop floor, that gap cannot simply be wished away.

The second approach is to create a "recovery slope" which accepts the cost increases but returns unit cost near to what it would have been had the extraordinary impact not occurred. This is clearly a more realistic approach than the first. But how quickly should we forecast recovery?

Frequently, an arbitrary number of units is chosen, and the recovery is then forecast over that number of units (See Figure 9). Sometimes, the choice of units is based on historical analogies. Sometimes it is based on a point in time when the manufacturing schedule recovers to the baseline. Sometimes it is simply picked out of the air. All these have problems. Our historical analogies may not be apt, or we may not have the data. Cost improvements usually lag schedule improvements, especially since schedule improvements are often made by increasing manpower or overtime or both. Bottom line, it is very easy to make unrealistic shop projections which cannot be achieved.

Solution: Calculating Learning Setback and Projecting Forward

A more reasonable approach is to take the break-in point of the disruption and set back the unit position on the learning curve (Fowlkes, 1963). For example, prior to the part shortage we were on an 85% slope. The first unit to feel the impact of the shortage represents approximately 850 hours per unit – equivalent to position 100 on that same 85% slope. To forecast the recovery, we regress on the learning curve back to unit 100 and forecast future units on the same pattern as established in the past, i.e., the next five units are equivalent to the cost of units 101 thru 105, etc. on an 85% improvement slope.

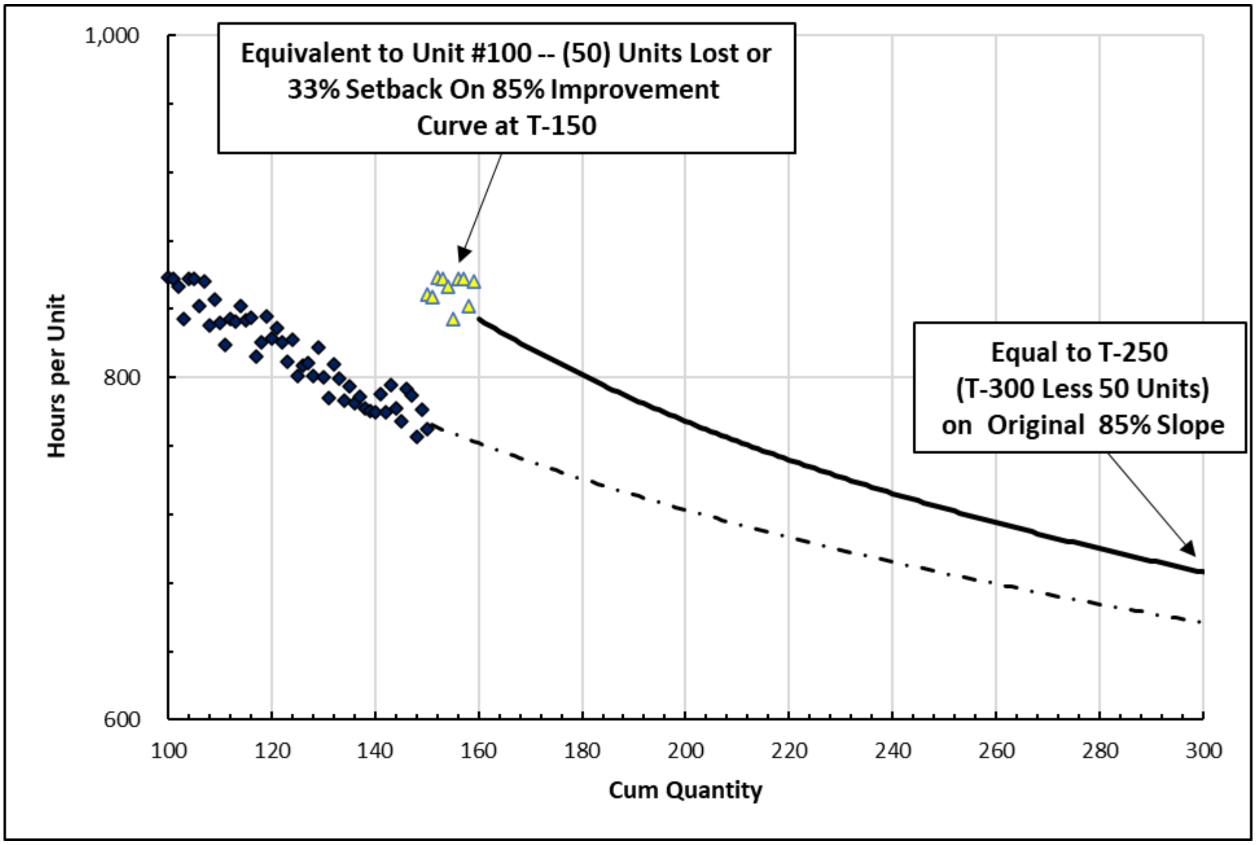


Figure 10. Recovery Curve – Setback

This produces a "scallop" in the overall improvement curve (See Figure 10). True to the learning curve pattern, the most improvement is seen in the initial units after the disruption, with the unit-to-unit decreases slowing as we move farther away from the initial disruption. In this case, we recover asymptotically to the old cost curve – that is, we never achieve the same hours per unit we would have anticipated had the disruption not occurred. But we come closer and closer to it until eventually the difference between the two becomes marginal.

The use of setback in the learning curve is widely accepted for production breaks and engineering changes (Anderlohr, 1968; DCAA, 1994; Smith, 1986). But there is sometimes resistance to using it in other scenarios.

This resistance is largely based around the idea that learning – and the loss of learning – exclusively centers around the operator on the shop floor. In the case of engineering changes (the operator must learn a new way of building the part) and production breaks (there is a significant turnover on the floor with employees receiving new assignments), there is clearly an impact to the body of knowledge the shop floor operator has accumulated. But our common use of the term "learning curve" often misleads us into believing that cost improvement only results from repetitive operations by the mechanics. It is more accurately called out as a "cost improvement curve."

In his paper on production breaks, Anderlohr defined five elements of learning: (1) operator learning, (2) supervisory learning, (3) tooling, (4) continuity of production and (5) manufacturing methods (Anderlohr, 1969). Yet the improvement that comes from the repetition of tasks by shop personnel accounts for slightly more than 20% of the total cost improvement. The rest of "learning"

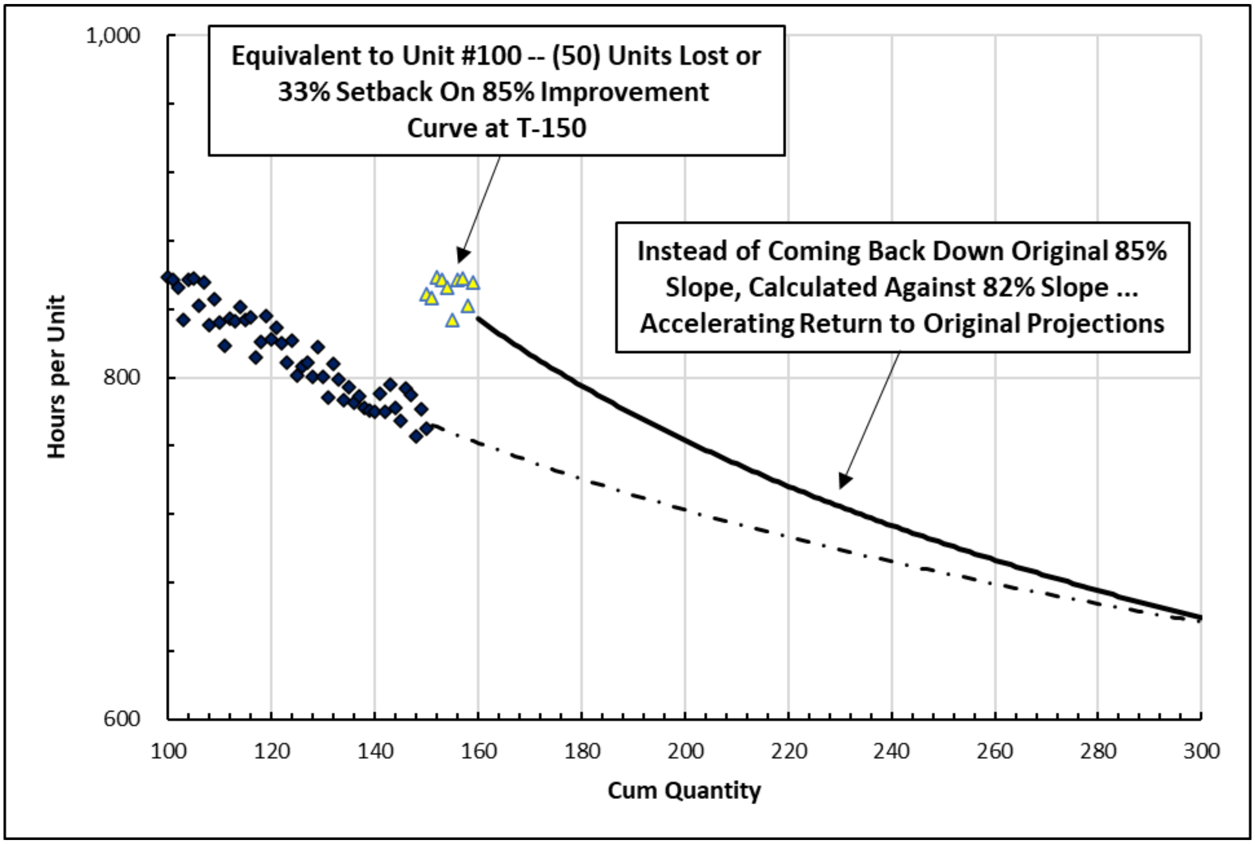


Figure 11. Recovery Slope – Accelerated Setback

comes from other sources. (Jones, 2001). If we can adjust our position on the improvement curve for negative impacts to operator or supervisor learning, surely it is legitimate to adjust it for negative impacts to the other three areas as well? Viewed from this perspective, it should be plain that, for example, an interruption to the supply chain due to late parts – which in turn creates part shortages, workarounds and behind schedule conditions – represents a retrograde to the existing improvement curve and can be fairly represented by a setback on the learning curve.

In the author's experience, this produces the most realistic and reasonable recovery slope and the one most achievable by the shop floor. But there are cases where a more aggressive approach may seem appropriate. Smith (1986) makes a common argument: "The firm is reexperiencing, not experiencing; they are going down a cost improvement curve they have been over before and should be better equipped to solve the problems the second time around so some method of accelerating recovery…may be useful". We can modify the setback methodology shown above and assume that we forecast the new units not on the same pattern seen in the past – the 85% improvement slope – but a slightly more aggressive one. In this case, an 82% slope has been used (See Figure 11).

This allows us to completely return to the hours per unit projected on the old cost curve. (In fact, had we continued the projection another ten or twenty units, the recovery slope would fall underneath the old cost curve, giving us a lower per unit value.) The more aggressive the slope assumption, the faster the interception point will be achieved. However, we can easily fall in the same trap as the earlier case where we selected an arbitrary number of units and drew a line to intercept the old cost curve. If 82% was an appropriate slope, why not 80%? Why not 78%? Why not 75%? It is easy to rationalize the answer

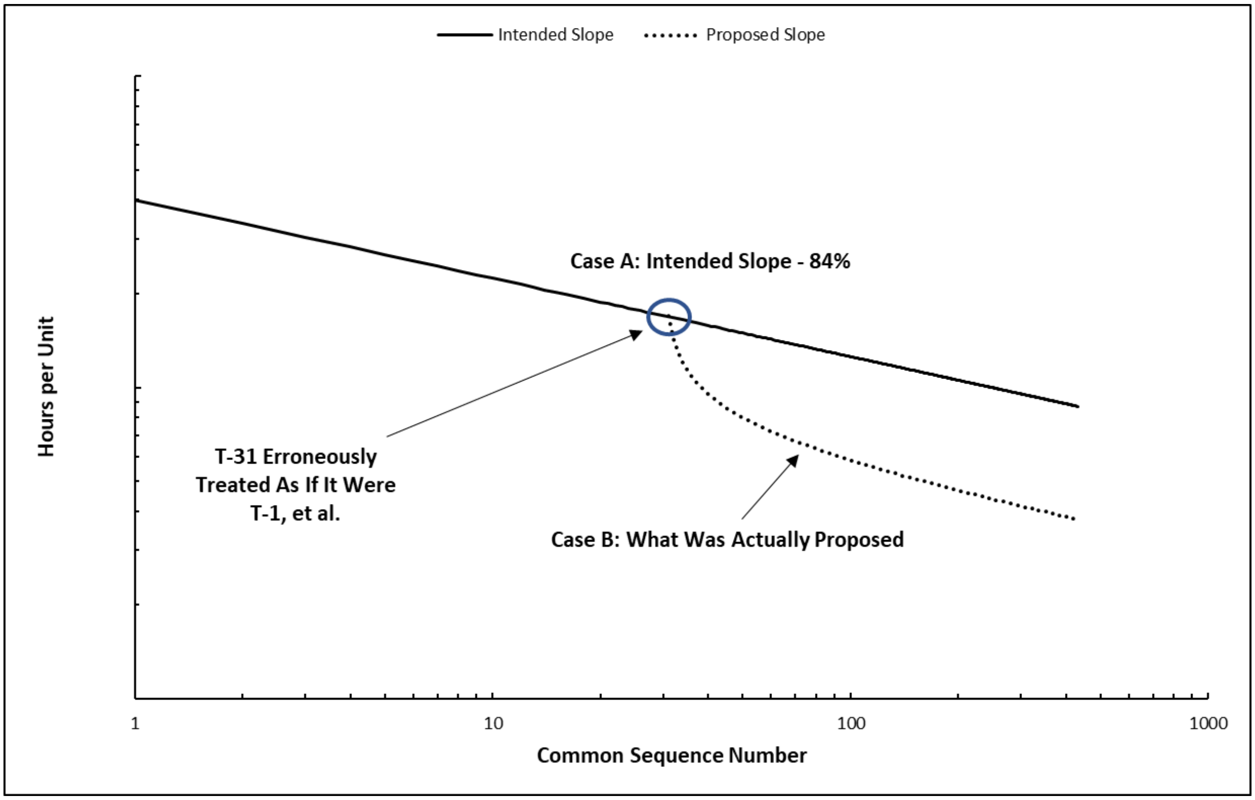


Figure 12. Illustration of Misidentified First Unit (Notional)

we (and our management) want to hear. Cochrane (1968) suggests a methodology for calculating an accelerated recovery, but it too requires an arbitrary choice of an acceleration factor which might be difficult to justify. The best guide would seem to be prior experience, but often we cannot find an analogy which exactly correlates to the situation we are estimating.

In short, projecting recovery slopes from disruptions is fraught with potential risks and the greatest care must be taken with doing so. When calculating a recovery slope, it is always best to review your assumptions and projections with the shop floor to make sure what you have mapped out can in fact be realized.

Peril: Being Careless When Establishing the First Unit

This example is drawn from an actual proposal. Values referenced below are notional.

A small aircraft pylon used to carry mission equipment required subassembly work. For the first thirty units, the Special Projects organization produced it on an 84% learning curve. At unit 31, the task was transferred from Special Projects to the regular Production department, who would produce the next order for 400 units.

The cost analyst (fortunately not the author!) proposed that the first Production unit would have the same hours per unit as the last unit produced by Special Projects. He also proposed the same 84% learning curve slope going forward. However, for projection purposes, he treated the first Production unit as unit one on the learning curve. The estimator apparently believed he was setting the unit costs back on the learning curve. But while he reset the cumulative unit count, he did not adjust the hours at unit #31 to something higher.

Figure 12 shows the consequences. Case A represents what the Production department expected to see when the contract was awarded.

Case B represents what they found in the estimate – an estimate that was approximately half of what they expected!

By treating T-31 and subsequent units as if we were restarting the learning curve back at unit 1, we have restarted the 16% cost reduction that occurs every time the number of units doubles. This significantly accelerates the rate of learning – which was not the intention of the estimator. GAO (2020) refers to this as "disjoint theory" (treating the first production unit as T-1 and restarting the curve) as opposed to "sequential theory" (treating the first production unit as the last development unit plus one). There may be times where disjoint theory is appropriate, but in this case the analyst simply did not realize what had been done.

Fortunately, Production was able to mitigate the impact by holding a series of lean events and substantially restructuring the production process – as it turned out, there were significant inefficiencies in the existing production process which were subsequently eliminated. However, this happy accident cannot be counted on in the future to save an estimator from his mistakes.

Solution: Take Care and Graph, Graph, Graph!

Fortunately, the solutions are relatively simple. As a rule, analysts should always graph their learning curve results – preferably in both a log-log and an arithmetic space. Graphing the actual cost history as well as the projected hours per unit would have quickly surfaced the problem. In addition, examine your takeoff point for projections and its position on the curve carefully. Seemingly insignificant decisions can have profound impacts on the numbers.

Peril: Steep Curves = Efficiency?

The author has heard proposal evaluators frequently assert that a flat learning curve is proof of manufacturing inefficiency. Its counterpart is often asserted as well: a steep learning curve proves the efficiency of a manufacturing operation. In fact, the slope of a learning curve by itself does not prove that a factory is efficient or inefficient. A hypothetical example will demonstrate this.

Company A assembles widgets; it has demonstrated an 80% learning curve over 1,000 units. Company B builds a similar but not identical product and demonstrates a 90% learning curve over the same range. There has been no transfer of manufacturing knowledge or personnel between the two companies. Which company is more efficient?

Many cost estimators would immediately answer Company A since it has the steeper learning curve. But this ignores the reasons why Company A had such a steep learning curve. This is quickly demonstrated by comparing the performance of the two companies on an hours per pound basis (See Figure 13). This shows Company A's high first unit cost, exceeding 40 hours per pound. Upon investigation, it turns out this high T-1 was driven by late engineering release, inadequate tooling, late material, and the oversizing of shop floor crews to recover manufacturing schedule.

Company B on the other hand had its engineering released on time, which allowed its tooling program to build high quality tools and deliver them to the floor when needed. On-time engineering allowed the supply base to deliver its parts on time, which in turn allowed Production to size its crews efficiently and still maintain the production schedule. Its first unit cost was almost half of Company A's.

Both companies ended the 1,000th unit at the same hours per pound. But over the course of those thousand units, it took Company A almost 25% more hours to produce its product.

A steep learning curve can demonstrate a strong dedication to lower costs and continuous improvement. It can also indicate the necessity to recover from poor performance and

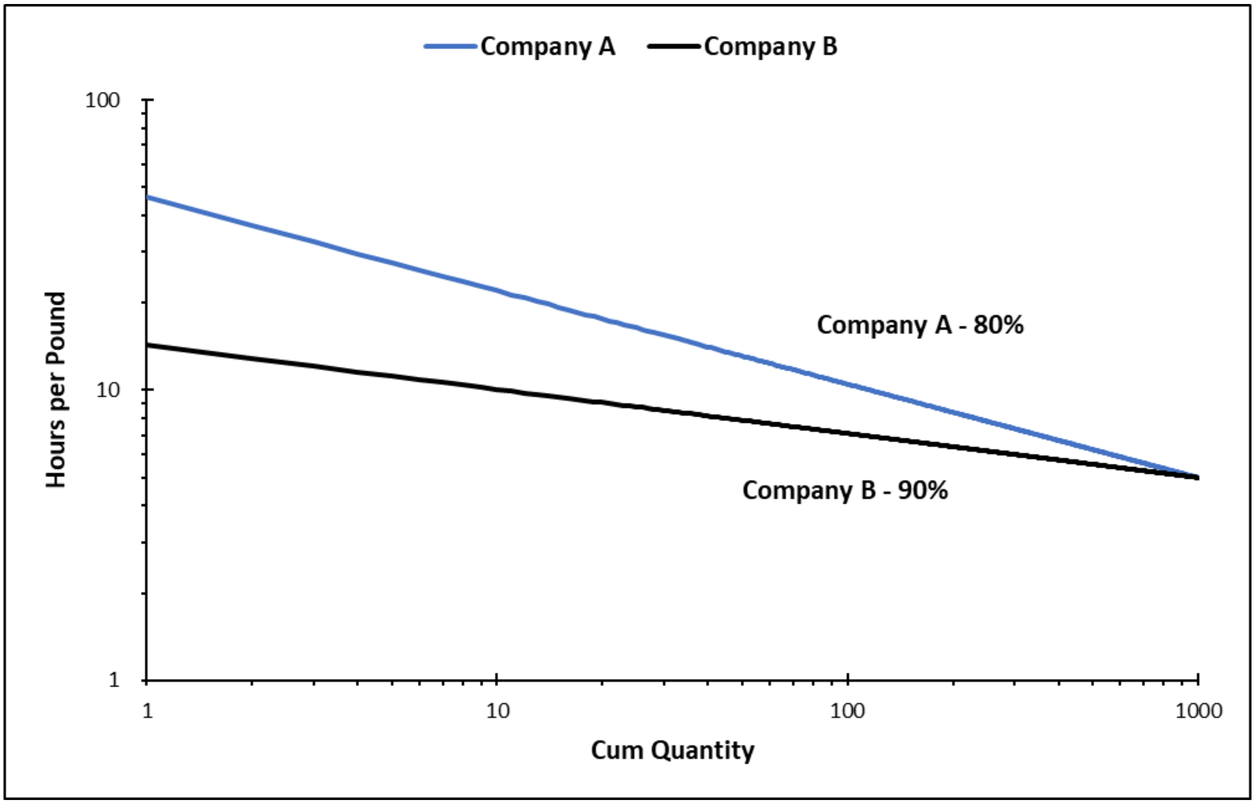


Figure 13

mismanagement on the earliest units. "[T]he more room there is for improvement," noted Fowlkes (1963), "the more improvement there is to be expected." Without further investigation, it cannot be determined from the numerical value of a learning curve slope alone which of these two cases is true.

Solution: Avoiding the Facile Conclusion

There is a widespread perception among cost estimators that relatively flat learning curves are a symptom of production inefficiency, and -- by implication – that relatively steep slopes are proof of manufacturing efficiency. In fact, as our example demonstrates, just the opposite may be the case. The learning curve slope alone cannot tell us if a manufacturing operation is efficient or not. Further analysis and understanding behind the underlying trends are necessary. Unfortunately, there is no easy way out.

Conclusions:

The quantity of books, articles, and academic research published about learning curves is astonishing. Literally hundreds of publications have been released since T. P. Wright's original 1936 article. Authors have suggested a variety of models and approaches: Wright's cumulative average model, Crawford's unit curve model, the Stanford B-curve, DeJong's incompressibility model and Cochran's S-curve are only a few examples. (Wright, 1936; Stanford Research Institute, 1949; DeJong, 1957; Cochran, 1960; Cochran, 1968) And yet within this wealth of material, there are relatively little guidance on what *not* to do. A new driver should not be handed a key to the sports car in the driveway before being previously schooled on speed limits and stop signs.

In the introduction, this paper was offered as a warning label to be attached to the improvement curve. Each of the five situations outlined represents a potential pitfall which can entangle the cost estimator and transform the learning curve from a useful tool to a danger to himself and others. The negative consequences of a bad estimate – on company profits and government funding – can be severe.

Unfortunately, most learning curve training rarely addresses these issues. It is content to show the basic calculations for Wright and Crawford curves, offer some advice on midpoint calculation and show a methodology for dealing with major engineering changes or production breaks. But it rarely goes much beyond these areas. It simply assumes estimators will find out about those other matters "soon enough." They will – but they might take someone else down with them in the process.

Cautionary tales rarely make compelling reading. After all, who among us actually reads the warning labels attached to the products we buy? But in this case, questioning long-held premises or putting in an extra half hour of analysis may yield unexpected benefits. By obeying the speed limits of estimating, our hypothetical driver and his sports car in the driveway might make it back home in one piece.

References

Anderlohr, George (1969, September). "What Production Breaks Cost," *Industrial Engineering*, 34-36.

Asher, H. (1956). *Cost-Quantity Relationships in the Airframe Industry*. Santa Monica, California: RAND Corporation.

Boone, E., Elshaw, J., Koschnick, C., Ritschel, J., Badiru, A. (2021). "A Learning Curve Model Accounting for the Flattening Effect in Production Cycles." *Defense Acquisition Research Journal*, 28(96), 72–97.

Carr, G.W. (1946, April). "Peacetime Cost Estimating Requires New Learning Curves." *Aviation*, *45*, 76-77.

Cochran, E.B. (1960, July-August). "New Concepts of the Learning Curve." *The Journal of Industrial Engineering*, 317-327.

Cochran, E.B. (1968). *Planning Production Costs: Using the Improvement Curve*. San Francisco, CA: Chandler Publishing Company.

Crawford, J. R. (1944). *Learning Curve, Ship Curve, Ratios, Related Data*. Burbank, CA: Lockheed Aircraft Corporation.

DCAA Contract Audit Manual (1996). DCAAM 7640.1, vol. 2. Washington, DC: Government Printing Office.

DeJong, J. R. (1957). "The Effects of Increasing Skill on Cycle Time and Its Consequences for Time Standards." *Ergonomics*, 1(1), 51-60.

Dutton, J., Thomas, A. (1984, April). "Treating Progress Functions As a Managerial Opportunity." *The Academy of Management Review*, 235-247.

Fowlkes, Tommie F. (1963) *Aircraft Cost Curves*. Fort Worth, TX: General Dynamics Convair Division.

Fox, B., Brancato, K., Alkire, B. (2008). *Guidelines and Metrics for Assessing Space System Cost Estimates*. Santa Monica, CA: RAND.

Government Accountability Office (GAO) (2020). *Cost Estimating and Assessment Guide: Best Practices for Developing and Managing Program Costs*. GAO-20-195G. Washington, DC: United States Government Accountability Office.

Hess, R. W., Romanoff, H.P. (1987). *Aircraft Airframe Cost Estimating Relationships: Bombers and Transports*. Santa Monica, CA: RAND.

Johnstone, Brent M. (2015) *Improvement Curves: An Early Production Methodology*. Paper presented at the 2015 International Cost Estimating and Analysis Association (ICEAA) Professional Development and Training Workshop, San Diego, CA. Retrieved from http://www.iceaaonline.com/ready/wp-content/uploads/2015/06/PA03-Paper-Johnstone-Improvment-Curves.pdf.

Jones, Alan R. (2001). "Case Study: Applying Learning Curves in Aircraft Production – Procedures and Experiences." *Maynard's Industrial Engineering Handbook*, 5th edition. New York, NY: McGraw-Hill.

Kennedy, Peter (1992). *A Guide to Econometrics*. Cambridge, MA: The MIT Press.

Large, Joseph P.; Hoffmayer, Karl; Kontrovich, Frank (1974). *Production Rate and Production Cost.* Santa Monica, CA: RAND.

Levinson, G. S.; Barro, S. M. (1966). *Cost Estimating Relationships for Aircraft Airframes*. Santa Monica, CA: RAND.

Miller, F. D. (1971, July). "The Cubic Learning Curve – A New Way to Estimate Production Costs." *Manufacturing Engineering & Management*, 14-15.

Nussbaum, Daniel A; Mislick, Gregory K. (2015). *Cost Estimation: Methods and Tools*. Hoboken, NJ: John Wiley & Sons, Inc.

Resetar, Susan A; Rogers, J. Curt; Hess, Ronald W. (1991). *Advanced Airframe Structural Materials: A Primer and Cost Estimating Methodology*. Santa Monica, CA: RAND.

Shumeli, Galit (2010). "To Explain or to Predict?" *Statistical Science*, 25(3), 289-310.

Smith, Larry L. (1986). *Cost Improvement Analysis*. Dayton, OH: Air Force Institute of Technology.

Stanford Research Institute (1949). *An Improved Rational and Mathematical Explanation of the Progress Curve in Airframe Production*. Stanford, CA: Stanford Research Institute.

Wright, T.P. (1936, February). "Factors Affecting the Cost of Airplanes." *Journal of the Aeronautical Sciences*, 3, 122-128.

Younossi, Obaid; Kennedy, Michael; Graser, John C. (2001). *Military Airframe Costs: The Effects of Advanced Materials and Manufacturing Processes*. Santa Monica, CA: RAND.

***Brent M. Johnstone** is a Technical Fellow and production air vehicle cost estimator at Lockheed Martin Aeronautics Company in Fort Worth, Texas. He has 34 years' experience in the military aircraft industry, including 31 years as a cost estimator. He has worked on the F-16 program and since 1997 has been the lead Production Operations cost estimator for the F-35 program. He has a Master of Science from Texas A&M University and a Bachelor of Arts from the University of Texas at Austin.*

(CE)^2 : Communication and Empowerment for Cost Estimators

Christina N. Snyder, CCEA®

The views and opinions expressed in this paper are those of the author alone and do not reflect the official policy or position of any other organization, employer, or company.

Abstract: Too often, cost estimator training focuses solely on technical abilities, largely ignoring the "soft skills". Good communication and empowering the team were shown in a 2020 ICEAA community survey to be critical to cost leadership efficacy. Knowing that technical skills alone will only take estimators so far, this paper leverages existing communication and empowerment best practices to demonstrate how all cost estimators can use soft skills to exponentially impact their analyses.

Acknowledgements: Thank you to the ICEAA community for giving an outlet to gain and share knowledge. Thank you to my husband, Dr. Benjamin Snyder, for not only allowing me search through all your old textbooks and journals but for also being a reviewer of this work. Thank you to Caleb and Griffin Snyder for allowing me to listen to all the "boring" leadership podcasts on the way to your practices.

Introduction

The International Cost Estimating and Analysis Association (ICEAA) is a cost estimating organization built by cost estimators for cost estimators. While having a technical-specific emphasis does facilitate state-of-the-art estimating training and incorporation of best practices in quantitative areas like data science, we sometimes lose focus on other opportunities for professional growth. As a community, we spend a lot of time building on our strengths, but this paper seeks to target a weakness for many technically competent professionals. Fortunately, there are many examples of anecdotal and empirical findings of the behaviors and attributes that lead to effective leadership in professional, educational, and popular media.

In a recent survey of cost analysts, (Snyder, 2021) the community unanimously agreed that leadership is important to the end cost product and that "soft skills" like being a good communicator are important to leadership efficacy. "Soft skills" are essentially people skills – and due to the nature of being less tangible, non-technical, and sometimes personality driven, they are harder to define and teach. These skills become increasingly important when one works on larger teams, begin leading cost projects, and/or as direct visibility to the decision maker increases. This paper expands upon previous work regarding soft skills to improve cost team outcomes such as: higher team morale, efficiently delegating work, and quickly identifying challenges. It seeks to leverage proven training on how to communicate and empower members of estimating teams more effectively. Although there is specific guidance to cost team leadership, given the nature of cost estimating, the insights can be helpful in all aspects and roles within our field. Much of the success of our work relies on efficiently receiving inputs from others and then communicating our results to decision makers. Taking personal inventory of strengths and limitations in soft skills and working on communication skills throughout the leadership chain will improve all estimators, not just those in cost lead positions.

EQ and Self-Awareness

"Leadership and learning are indispensable to each other"
– John F. Kennedy

Effective leaders boost team morale, create strong relationships in the workplace, and help others embrace stretch goals; all of these key skills require emotional intelligence (EQ).[1] There are several different interpretations of EQ (emotional quotient). For the purposes of this paper, we will define EQ as the ability to recognize your own emotions, knowing how you are perceived by others, understanding your impact on those around you, and being able to use that knowledge to motivate or adapt the behaviors. To bring out the best in others, it is critical to first understand your own strengths and weaknesses through self-awareness. Personality and EQ assessments seek to measure one or more of the following: personality traits, dynamic motivation, symptoms of distress, personal strengths, and attitudinal characteristics.[2] The path to self-awareness involves honest communication with yourself and those that know you best. This communication can be complimented by structured self-report personality and EQ assessments readily available on the internet.

Dr. Tasha Eurich published the chart above in the Harvard Business Review that outlines internal and external self-awareness3. Dr. Eurich distinguishes internal self-awareness as how clearly we see ourselves, our aspirations, our strengths/weaknesses, and our impact on others. External self-awareness is how other people view us for those same criteria. Research has shown there is no relationship between internal and external self-awareness,3 so it is possible to rate high on one scale and low on the other. It is critical that both new and veteran leaders take time to focus on self-awareness as it is positively associated with important leadership outcomes. In a study of more than 1,200 leaders in a variety of industries, higher level leaders had larger discrepancies between their self-assessments and the assessment of them by those around them; meaning the larger a leader's team, the more important it is to close that gap by improving the communication with those that report to them.[4]

The Four Self-Awareness Archetypes

This 2x2 maps internal self-awareness (how well you know yourself) against external self-awareness (how well you understand how others see you).

	Low external self-awareness	High external self-awareness
High internal self-awareness	**INTROSPECTORS** They're clear on who they are but don't challenge their own views or search for blind spots by getting feedback from others. This can harm their relationships and limit their success.	**AWARE** They know who they are, what they want to accomplish, and seek out and value others' opinions. This is where leaders begin to fully realize the true benefits of self-awareness.
Low internal self-awareness	**SEEKERS** They don't yet know who they are, what they stand for, or how their teams see them. As a result, they might feel stuck or frustrated with their performance and relationships.	**PLEASERS** They can be so focused on appearing a certain way to others that they could be overlooking what matters to them. Over time, they tend to make choices that aren't in service of their own success and fulfillment.

SOURCE DR. TASHA EURICH

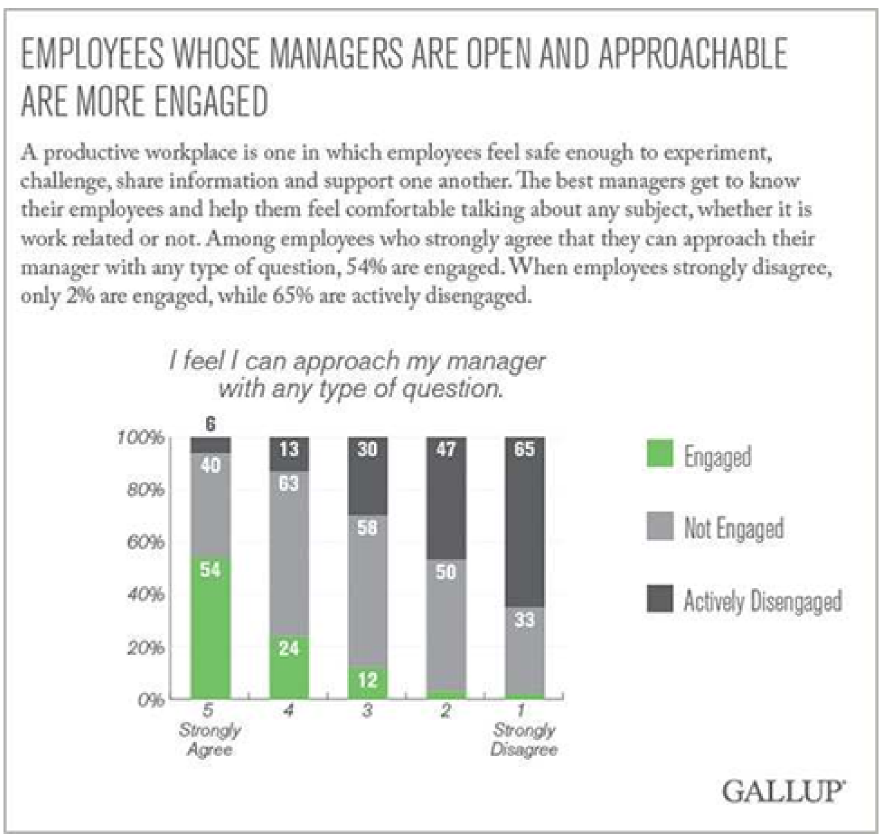
Communication

"Communication is the real work of leadership."
-Nitin Nohria, Dean of Harvard Business School from 2010-2020

Part of what differentiates humans from the rest of the animal world is our ability to work as a team through our communication. Unfortunately for our profession, most cost team leads are put into those positions solely due to their technical abilities. However, high technical achievement does not necessarily teach the skills or empathy needed to understand other people. When leading an estimating team, leaders should be looking for opportunities to increase productivity, maximize employee engagement, and effectively navigate the estimating process. All of these can happen by improving team communication. Using best practices across multiple industries, we can get inspiration for the best communication strategies leaders can implement to impact our cost estimating teams. It should be noted that the literature on leadership say the best leaders continually develop leaders within their own team, so all members of a cost team can benefit from beginning to practice these communication strategies at any stage in their career. The communication skills found most often to be employed by effective leaders were: adaptability, approachability, trust, and inspiration.

Adaptability – Adaptability in leadership involves understanding and accommodating the communication styles of each team member. The best leaders use their EQ to understand that each person is different and one of the keys to effectively communicating is tailoring communication styles, both verbal and nonverbal, to the audience. Such tailoring allows you to be more persuasive and build stronger relationships with your team members. Some team members may appreciate short emails with no pleasantries while another may interpret that style as cold or harsh. Leaders should give their teams hope and support, so being careful with words and having a cool head in stressful situations might be especially important. Ultimately, using tact and showing respect when interacting with every team member sets the foundation for modeling a collaborative and efficient work environment. Thus, having self-awareness regarding your default communication style and adapting to your team's individual needs is critical.

Approachability - Most of us learned as children that we have "two ears and one mouth, because we should listen twice as much as we talk." Active



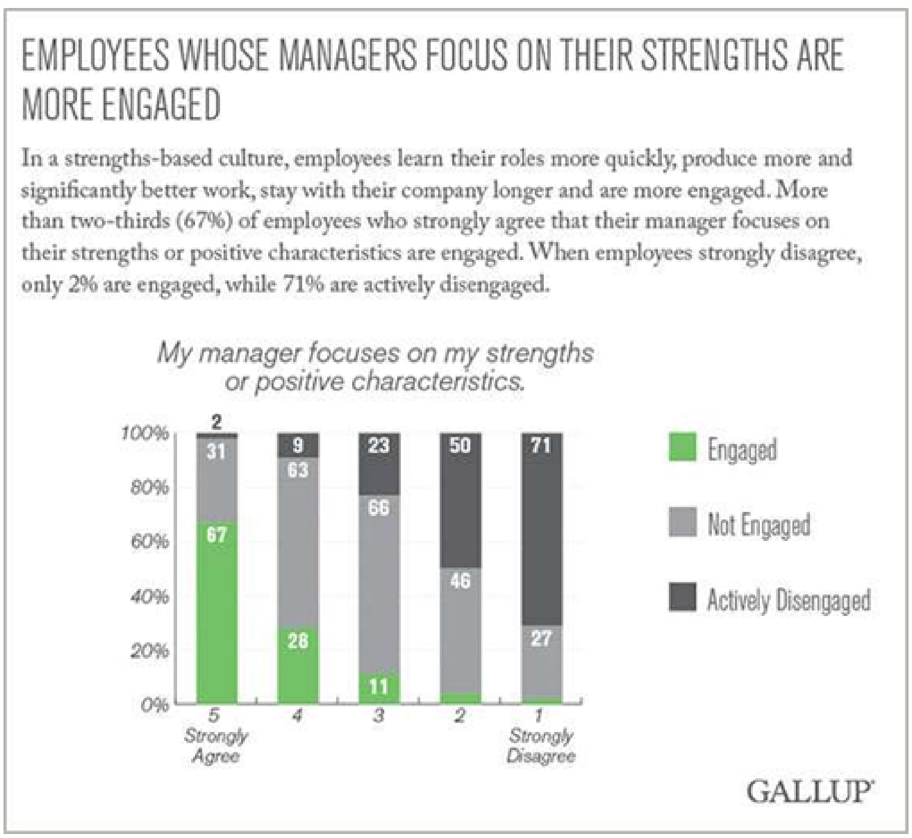
listening should be the foundation of communication for every leader. If team members know that you are really listening to their inputs, they are more apt to openly share their ideas and provide honest feedback. As the Gallup poll shows,6 this drives employee engagement, but also increases productivity and creativity. Approachability means setting time aside to speak directly to your team and removing distractions like not looking at your phone or computer screen when they speak. When a person brings issues to your attention, empathize with their feelings appropriately, and encourage them to continue talking.

Trust - Consistent in the literature on team effectiveness, trust has been shown to be the most important component of any successful team. When teammates fear each other, fear their leader, or believe they are in competition, they lose the ability to be creative for fear of making mistakes and will hoard knowledge and resources to give themselves a tactical advantage over other team members. To have team members trust each other, the leader needs to model transparency, integrity, and vulnerability through consistent action. The book "Connect – Building Exceptional Relationships with Family, Friends, and Colleagues" outlines the six hallmarks of exceptional relationships that stem from these areas of trust and authenticity:[7]

1. You can be more fully yourself, and so can the other person.
2. Both of you are willing to be vulnerable.
3. You trust that self-disclosures will not be used against you.
4. You can be honest with each other.
5. You deal with conflict productively.
6. Both of you are committed to each other's growth and development.

If you can build trust in you as the lead and within the team dynamic, all members will be more willing to share ideas and learn.

Inspiration – A leader that believes in their people can have a positive effect on the overall culture of their team. This effect involves celebrating success, framing missteps as opportunities for growth, and providing more positive than negative feedback. A 2004 study[8] showed leaders should be in the habit of providing significantly more positive feedback than most would assume. This research showed



high performing business teams had a positive to negative comment ratio of 5.625, medium performing teams had a ratio of 1.875, and the low performing teams had a positive to negative comment ratio of .365.[8] When placed in a bad team culture, even a star employee can become apathetic and perform to less than their abilities. Conversely, wanting to meet the expectations of an inspiring leader that believes in you can cause a mediocre performer to achieve more than they themselves felt possible. Leadership inspiration has resulted in greater productivity, greater workplace satisfaction; and employee engagement has been shown by Gallup to be superior to the more authoritarian approach. Procter & Gamble runs a development program which focuses explicitly on teaching leaders on how to inspire colleagues; A.G. Lafley even goes as far as to say, "The command-and-control model of leadership just won't work 99 percent of the time."[9]

Empowerment

"As we look ahead into the next century, leaders will be those who empower others"- Bill Gates

To paraphrase from the Google re:Work website, the most effective leaders usually realize they work for their teams and not the other way around.[10] Part of leading others is believing your team can become the best version of themselves and make an important contribution to the group effort. When developing cost estimates with a team, it is a balancing act for a cost team lead to figure out how to best utilize the available resources to complete the workload in a timely manner with the highest quality work. Taking into consideration the skillset and past performance of the individual, a cost leader can set high but achievable goals. True leaders don't manage or mandate actions or tasks; instead, they motivate and empower staff to identify and complete the work necessary for the established outcome.[11] The best way for a leader to do this is the commit to planning up front; the more work completed before the project begins, the clearer expectations and trust. As Dave Stachowiak, EdD puts it, "by doing more work up front, we minimize the amount of rework, conflict, and unclear expectations." By putting into practice the seven steps of delegation set up by Dr. Stachowiak, a cost team leader can ensure the whole team feels empowered and that communication is happening throughout the project.

1. *What does success look like?* Begin with the end in mind, envisioning the expectations for the timeline and final delivered product. Plan with as much detail as possible the expected outcomes for the final deliverable(s): What will be in the scope of the work; What will the documentation look like; Who will receive the briefing. This planning process needs to be completed by the team lead alone before any tasks are delegated.

2. *Who is the right person for this particular job?* The size of the team and capabilities within available time/resources need to be taken into consideration. You may have one person that is your go-to, but it can be demotivating if this person feels disproportionately over-burdened. Think of the capability and motivation of the team members. Each person on the team is not just doing a job but could also use the existing work as an opportunity to stretch abilities, build leadership confidence and skills beyond their current capabilities.

3. *Communicate expectations.* Take time to clearly set expectations for which team members will be responsible for sections of the work and try to be as detailed as possible. Keep in mind the communication principles discussed earlier of adapting your style to individual members and inspire them by reiterating that you believe they can accomplish the work.

4. *Individual estimator (staff member) plans the project.* Once individual responsibilities

within the estimate are clarified, allow the estimators to plan out the details of their own work. Not only does this convey trust but it allows them to have ownership over the work that they are doing. While considering previous performances, assign appropriate levels of autonomy to keep them engaged, allowing every estimator on the team, from juniors to seniors, to be creative and bring their unique ideas to the team.

5. Review the plan with the individual estimators (staff member). If there are any gaps that you think will affect quality, objectives, or timeline on the final deliverable, make sure to address them up front. However, resist the temptation to change inconsequential things that you may have done differently to maximize each member's feelings of empowerment and ownership.

6. Set the check-in drumbeat. At the outset, determine clear milestones and expectations regarding follow-ups. This clarity will provide an effective way for leaders to provide direct feedback without being perceived as micromanaging. It is important, as the team lead, to realize that no feedback is worse than negative feedback

7. Provide access to resources. If they need to get information from external stakeholders make sure those connections have been made up -front and that data requests or information will be forthcoming. Help people anticipate obstacles so they can figure out how to work through them. Even if you do not 1have full answers for all challenges or questions, be honest and forthcoming to let them know if there are problems that need to be worked through.

8. *Empower your team.* Empowering your team has an exponential impact. Not only will your team feel more ownership over their work and inspired by your trust, but leadership research shows that employees who feel more empowered are more likely to have higher levels of trust in their leaders.

Summary

"Leadership is about making others better as a result of your presence and making sure that impact lasts in your absence."

ICEAA members and the cost community unanimously agreed that leadership is important to the end cost product and that "soft skills," like being a good communicator and an empowered team, are important to leadership efficacy. By working on being highly internally and externally self-aware, focusing on effective communication, and efficiently delegating work, cost team outcomes – including estimate development time and efficiencies in the process – will also improve. Much of the success in cost estimating relies on efficiently receiving inputs from others and then communicating our results to decision makers. Taking an inventory of your own strengths and limitations of your own soft skills and working on your communication skills at every level of leadership will improve all estimators.

References

1. Colvin, G. (2019). Applying the Principles in Our Organizations. In Talent is overrated: What really separates world-class performers from everybody else (pp. 126–144). essay, Nicholas Brealey.

2. Gregory, R. J. (2016). Origins of Personality Testing. In Psychological testing: History, principles, and applications (pp. 314–315). essay, Pearson.

3. Eurich, T. (2018, January 4). What self-awareness really is (and how to cultivate it). Harvard Business Review. Retrieved February 22, 2022, from https://hbr.org/2018/01/what-self-awareness-really-is-and-how-to-cultivate-it

4. Sala, F. (2003). Executive blind spots: Discrepancies between self- and other-ratings. Consulting Psychology Journal: Practice and Research, 55(4), 222–229. https://doi.org/10.1037/1061-4087.55.4.222

5. de Vries, R. E., Bakker-Pieper, A., & Oostenveld, W. (2010, September). Leadership = communication? the relations of leaders' communication styles with leadership styles, knowledge sharing and leadership outcomes. Journal of business and psychology. Retrieved February 21, 2022, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2912722/

6. Adkins, J. H. and A. (2021, November 20). Employees want a lot more from their managers. Gallup.com. Retrieved February 22, 2022, from https://www.gallup.com/workplace/236570/employees-lot-managers.aspx

7. BRADFORD, D. A. V. I. D. (2021). Connect: Building exceptional relationships with friends, family, and colleagues. DOUBLEDAY.

8. Losada, M., & Heaphy, E. (2004). The role of positivity and connectivity in the performance of Business Teams. American Behavioral Scientist, 47(6), 740–765. https://doi.org/10.1177/0002764203260208

9. Colvin, G. (2019). Talent is overrated: What really separates world-class performers from everybody else. Nicholas Brealey.

10. Google. (n.d.). Re:work - guide: Empower your team. Google. Retrieved February 21, 2022, from https://rework.withgoogle.com/guides/managers-empower-your-team/steps/introduction/

11. Stachowiak, Dave. (n.d.). 7 steps of delegation - coaching for leaders. Retrieved February 21, 2022, from https://coachingforleaders.com/wp-content/uploads/2014/07/7delegationsteps.pdf

12. Bojeun, M. C. (2014). Program management leadership: Creating successful team dynamics. Auerbach.

***Christina Snyder** has been a cost estimator for 17 years; supporting multiple DOD programs with various aspects of cost estimating and strategic planning. She received her CCE/A® certification in 2012 and is passionate about translating cost data into insights that resonate. She won a best paper award in 2021 for her original work, "Dost Cost Team Leadership Matter". Mrs. Snyder graduated from Virginia Tech in 2005 with a B.S. in Applied Computational Mathematics.*

Theory of Complex Work

Harry T. Larsen

Background

The manufacturing learning curve graph depicts the direct labor hours per unit of production plotted in its manufacturing sequence. When plotted with log/log scales, learning curves tend to follow a linear downward trend as shown in figure 1.

The power law, $\mathbf{y(x) = a\, x^b}$, has been found to describe the learning curve's trend, with $\mathbf{y(x)}$ = hours(unit), $\mathbf{x}$ = sequential unit number, $\mathbf{a}$ = the hours at unit one, and $\mathbf{b}$ the slope exponent. The power law is transformed to a linear form by taking its logarithm: $\ln(\mathbf{y}) = \ln(\mathbf{a}) + \mathbf{b} \ln(\mathbf{x})$. To summarize a learning curve's statistics, a least squares regression can be performed on the logarithms of learning curve data, $(\ln(y_1), \ln(1))$; $(\ln(y_2), \ln(2))$; $\cdots$, $(\ln(y_n), \ln(n))$. The regression estimate of $\mathbf{a}$ is termed the theoretical number one. By convention, the slope of a learning curve is $2^{\mathbf{b}}$. The learning curve is sometimes transformed to a cumulative average:

$$y_{cum}(x) = \sum_{i=1}^{x} y(i) / x$$

For this discussion, the hours per unit description will be used.

From an examination of historic learning curve's labor hours in the airframe industry, some attributes are apparent.

The slope of a long sequence of end item's labor hours is very seldom steeper than 70.7%, $\mathbf{b}$= -.5.

When new work, e.g., a new design, for part of an end item is introduced at a unit, the hours for the new work are approximated by $\mathbf{y(x) = a_{new}\, x^b}$, with $\mathbf{x}$ beginning at unit 1. The replaced work is removed via $\mathbf{y(x) = a_{replace}\, x^b}$, with $\mathbf{x}$ at its current value. With $\mathbf{a_{new}}$ = #1 hours for the new work and $\mathbf{a_{replace}}$ = the #1 hours for the replaced work. This produces a spike in the learning curve.

Processes earlier in the manufacturing process have flatter slopes than do the later activities.

In the aircraft industry major and final assembly curves often steepen as they progress, while fabrication curves typically are less steep.

The learning curves of very large quantities of end items flatten.

The learning curves of products with large aggregates of hours per unit tend to be smoother than learning curves of fewer hours per unit.

The probability distributions about regressions of power law to labor hour data are approximately log-normal. (This is true for both learning curve hours per unit as well as Cost Estimating Relationships (CERs) for labor hours.)

There is a large cost variation around estimates for complex projects, particularly those that involve new technology.

Although the power law describes broadly the learning curve's log-linear behavior, it does not offer insight into the causes or mechanisms of its realization. Nor does it suggest means by which its labor hours per unit may be affected .

Theory of Complex Work

This model of work is based on the concept of a task.

The construction of a product is accomplished by the successful completion of a set of tasks.

A task is an activity that has a criterion of success.

A task is accomplished through repetitive trials.

The trials are repeated until the criterion of success is met.

A trial is modeled as a binomial event, with a probability of success **p(t)**, with **t =** trial.

At the completion of each trial, **p(t),** is increased by an amount **(1 - p(t)) dl,** where **dl** is a constant and **dl** << **p(t)**.

p(t+1) = p(t) + (1 - p(t)) dl

The time duration of each trial is a constant.

Thus, a task's labor hours are proportional to the number of trials necessary to successfully complete the task. A task may be a part of a larger interrelated group of tasks. The outcome of a task's successful trial may cause another task's success criterion to change, requiring it to be redone. The successful completion of a product's tasks produces a unit of the learning curve.

To model this process, at each trial, **p(t)** is compared to **x ~** U(0,1), an event from a uniform distribution. If **x** < **p(t)** the trial is a success and the task for that unit is completed, otherwise the trial is repeated. Trials for the next unit's task begin with the **p(t)** from the preceding completed task. The sequential completion of tasks produces the learning curve.

This is a stochastic process, with inherent uncertainty. To produce an estimate, it is implemented as a Monte Carlo system, producing Probability Density Functions, PDFs. From those PDFs various statistics can be calculated, e.g., median, mean, and standard deviation. Or, if the estimate is to support a decision option, the PDF can be bifurcated.

Fig 1 depicts a learning curve created by such a sequential completion of tasks. Its initial probability, **p(0)**, is .03 while **dl** equals .001, with 400 tasks and a trial time of 1 hour.

Predictions

The expected value of the number of trials to a successful completion is approximately 1/**p(t)**. Thus, with **p(t)** small, the change in **p** between

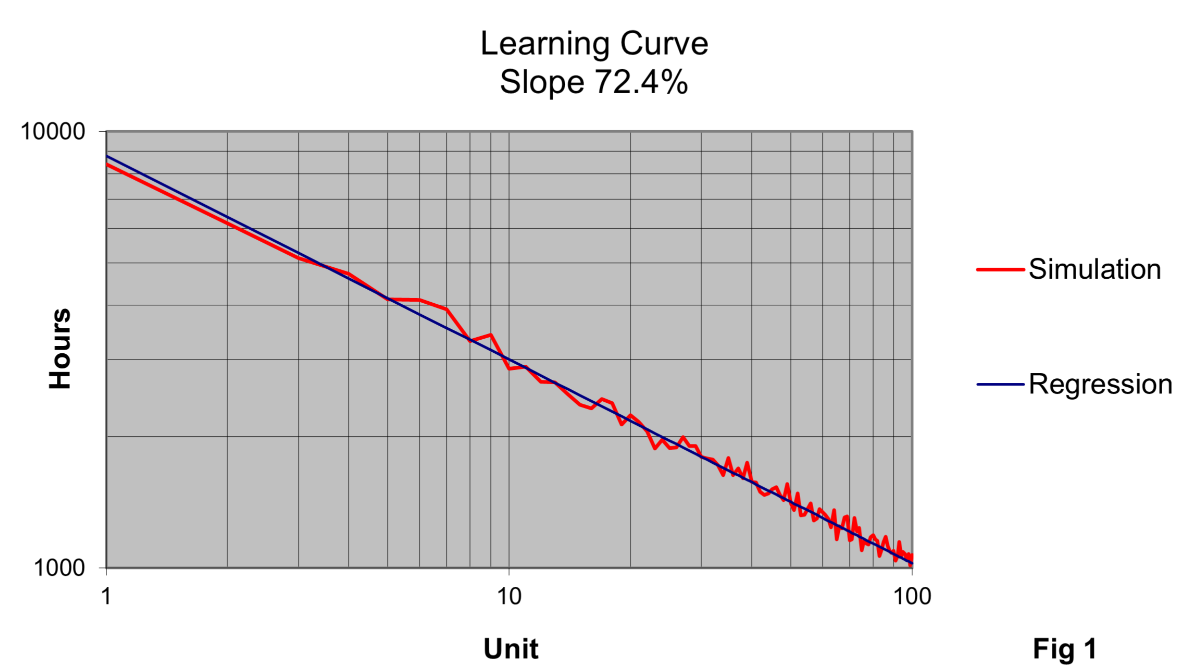


Fig 1

successful trials is about **dl/p(t)**. If this value is small the number of trials to successfully complete a set of tasks is a negative binomial distribution. A negative binomial distribution typically is defined as the discrete probability distribution of the count of failed trials up to a successful trial in a set of experiments. However, for this application, the count includes the successful trial.

The mean of a negative binomial distribution is:

(1) **u = T N / p**, where **N** is the number of tasks, and **T** is the time per trial for a task.

Its standard deviation:

(2) $\sigma = T (N (1-p))^{.5} / p$,

The relative standard deviation is:

$$\sigma / u = (T (N (1-p))^{.5} / p) / (T N / p)$$

or

(3) $\sigma / u = ((1-p) / N)^{.5}$

The number of tasks is: From (3)

(4) $N = (1-p) / (\sigma/u)^2$

From equation (1) it can be seen that a project's expected hours are proportional to the size of the project in terms of the number tasks, **N**, trial duration, **T**, and the difficulty of the project measured by the inverse of a trial's probability of success, **p.** Thus, a project may have high labor hours due to either its task content or its difficulty.

From equation (2), for **p** << 1, the standard deviation of labor hours is proportional to the square root of the number of tasks and the inverse of a trial's probability of success, 1/ **p**. Consequently, as illustrated by the relative standard deviation formula (3), a project with a higher task content, while holding **p** constant, will have a relatively lower standard deviation. Hence its depiction on log/log scales becomes smoother as the project's task content increases. Conversely, for a project that increases in size is due to increased difficulty (smaller **p**), its sequential standard deviation remains directly proportional to the increased labor hours and does not collapse with the larger project size. Thus, we should expect that large projects that advance the state of the art will have high relative standard deviations, while projects that do not will have low relative standard deviations.

For small **p** the relative standard deviation, **σ/u,** is a function of **N.** Figure 2 depicts two learning curves, one with 200 tasks and one with 40 tasks. The relative standard deviation of the 40 task curve is $(200/40)^{.5}$ times the 200 task curve.

For very large quantities a learning curve may flatten. When **p** approaches 1 the hours per unit

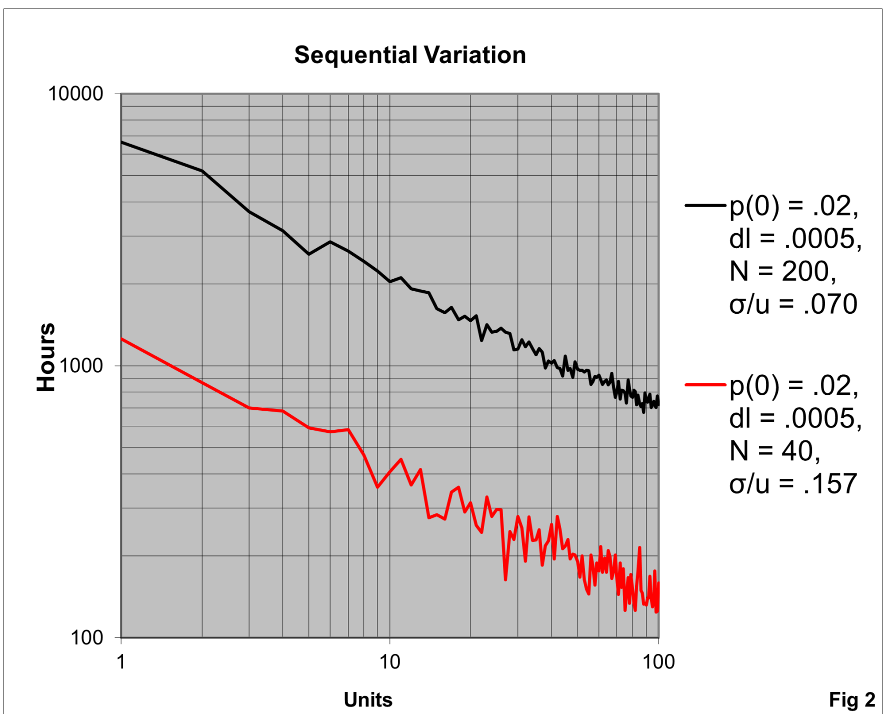


Fig 2

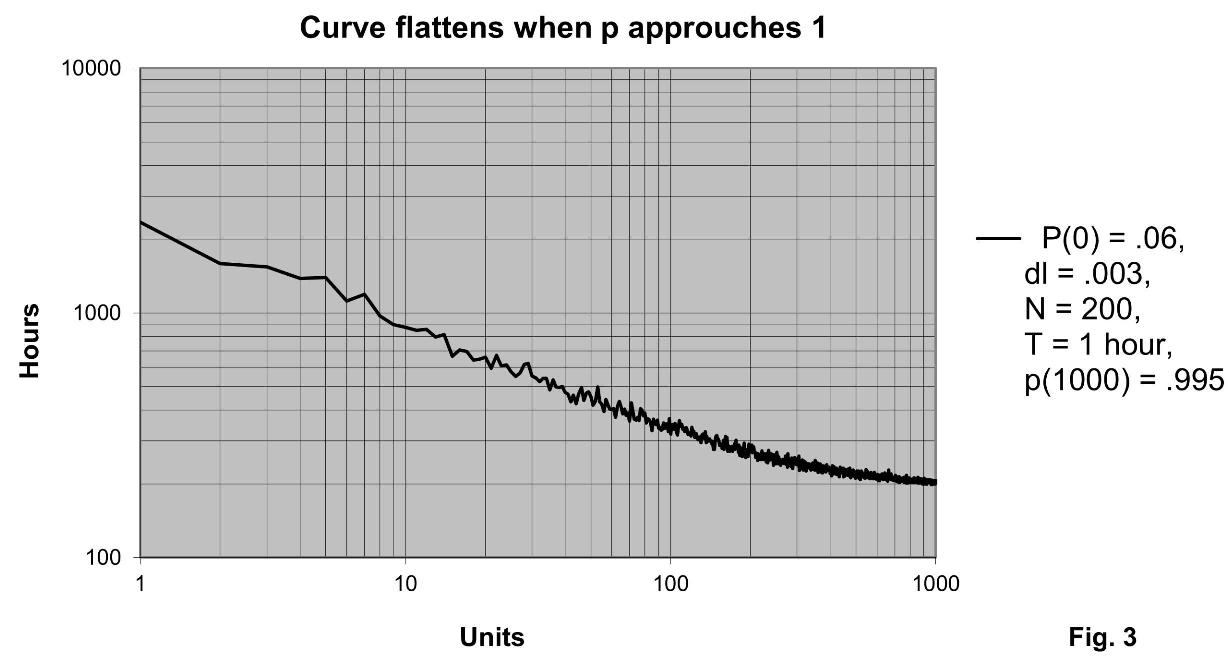


Fig. 3

approaches the product of the number of tasks and the hours per trial.

Figure 4 illustrates that **p(0)** and **dl** determine the hours for the first unit. As **p(0)** is increased the learning curve has a lower initial cost but remains asymptotic to a slope of 70.7% until **p(t)** approaches 1.

A combination of learning curves with differing **p (0)** values can produce a curve with a shallower slope.

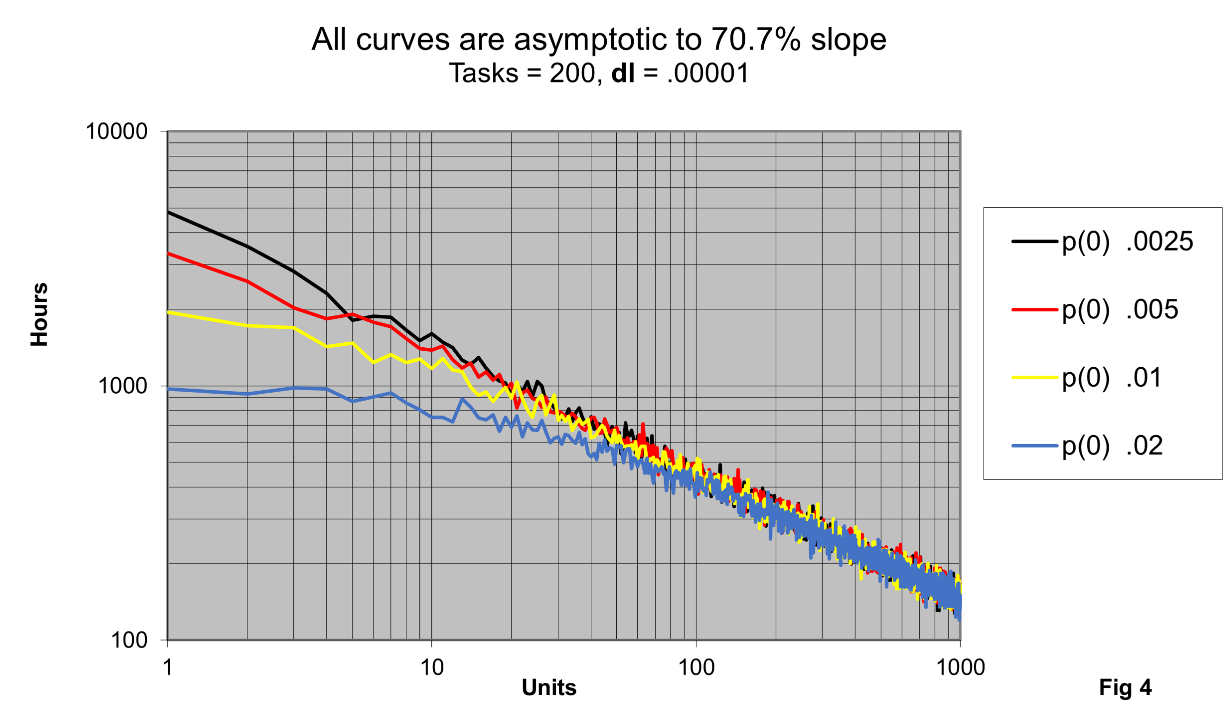


Fig 4

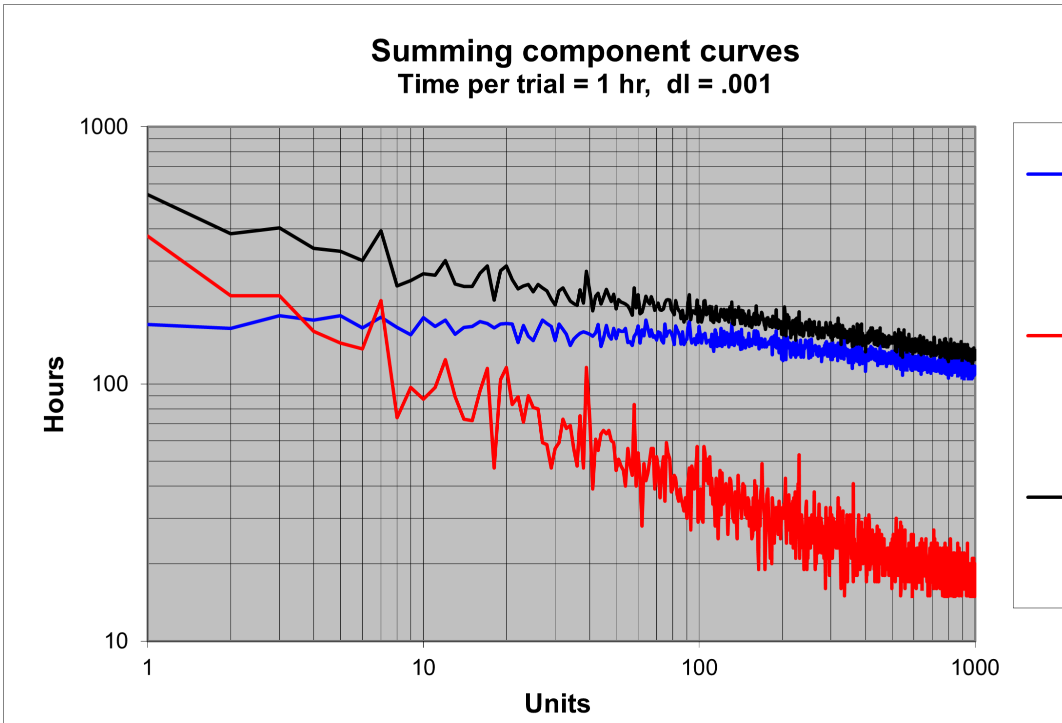


Fig 5

More realistically, a distribution of **p(0)** values can produce a learning curve with a small hump followed by a flattening. Figure 6 is a histogram of a lognormal distribution of 200 **p(0)** values, with a mean of .233 and a standard deviation of .133. Figure 7 is the resulting learning curve of 200 corresponding tasks beginning with those p(0) values. A log-log regression through unit 500 of the 1000 units shown has a slope of .822, a slope in the range commonly seen in aircraft learning curve history.

If a design change is introduced to an existing production process, it is added at the initial probability of the task, **p(0).** The work replaced is removed at its current probability, **p(x).**

In figure 8 the yellow line represents 70% of the tasks, which are unchanged. The red line scaled from one shows the new tasks. The orange, beginning at unit 200, shows the sum of new tasks and the continued unchanged tasks.

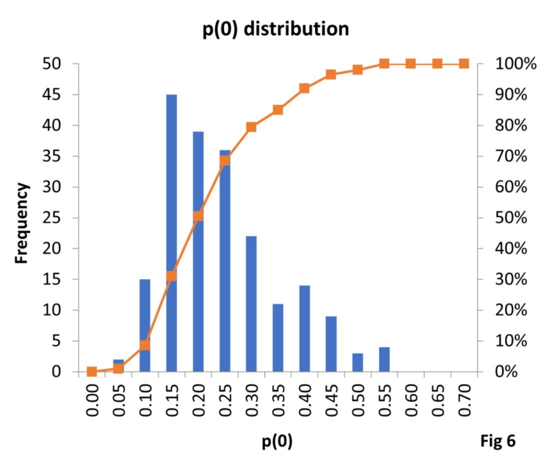


Fig 6

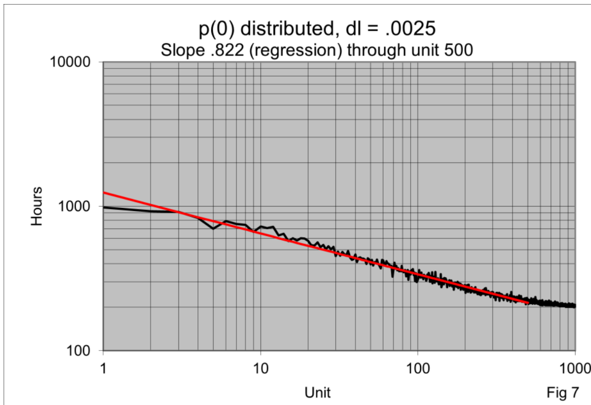


Fig 7

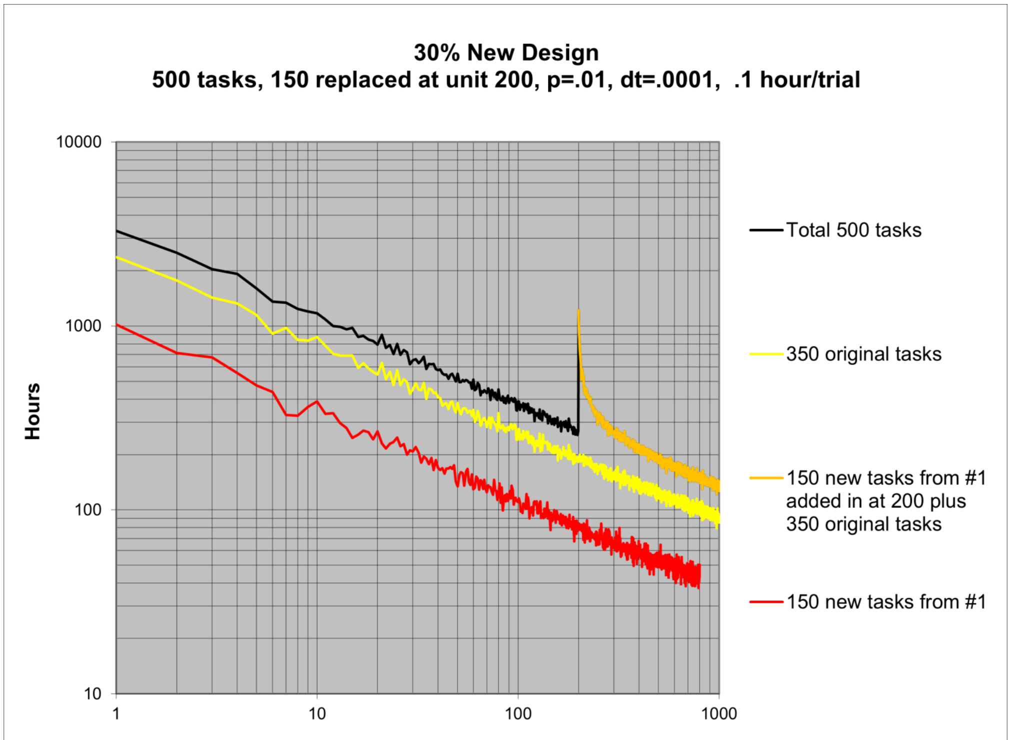


Fig 8

Feedback may occur between its tasks. When, for example, in an assembly task if a part does not fit due to a design or manufacturing error, the part may be reworked in the assembly activity to fit, but also, a design or specification change may be fed back to the fabrication area. The design change is treated as a new design, setting that task's current **p** to **p(0)** as shown in figure 6, but for a single task. Figure 9 shows a simulation of two sets of 150 tasks, fabrication and assembly. Design changes are fed back from assembly to fabrication. In this example, the likelihood of a fabrication task's **p** being reset to **p(0)** is proportional to the product of .0015 and the ratio of the number of trials for the preceding unit in assembly versus those in fabrication. Without

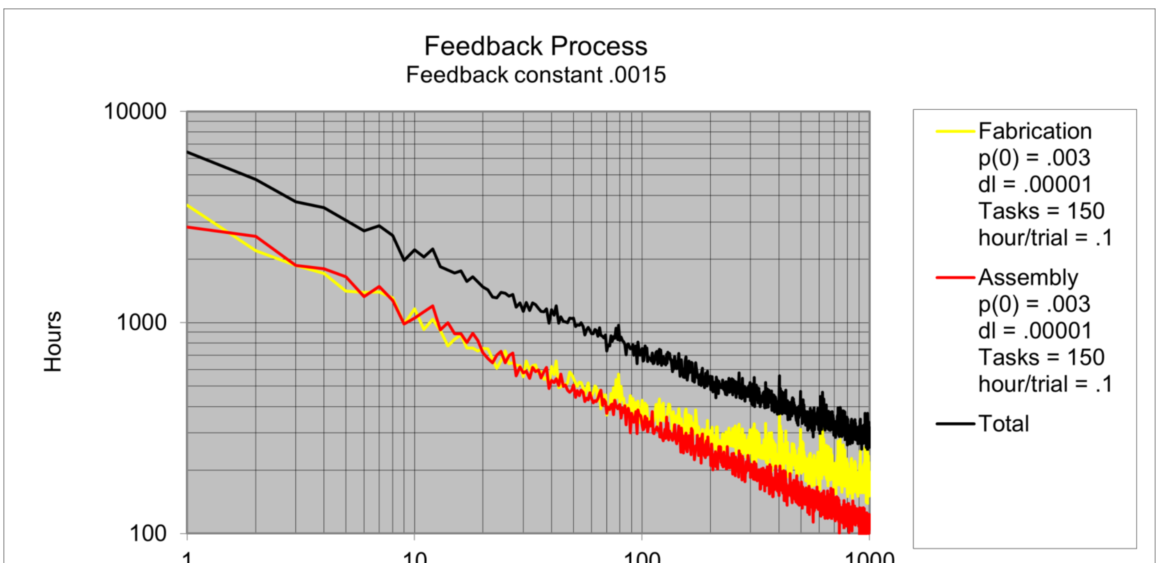


Fig 9

feedback, the fabrication curve would follow the trend of the assembly curve.

The Derivation of Model Parameters

For a system that cannot be transformed into a form suitable for the application of a regression, the author would normally use Excel Solver to find the least squares solution. Such a solution requires that gradients be calculated for each parameter at each step of the search. For a Monte Carlo system producing a partially random output a very large number of model executions would be required. Probably not computationally feasible, at least on a personal computer. Fortunately, a deterministic formulation of the model that very closely approximates the stochastic version is possible.

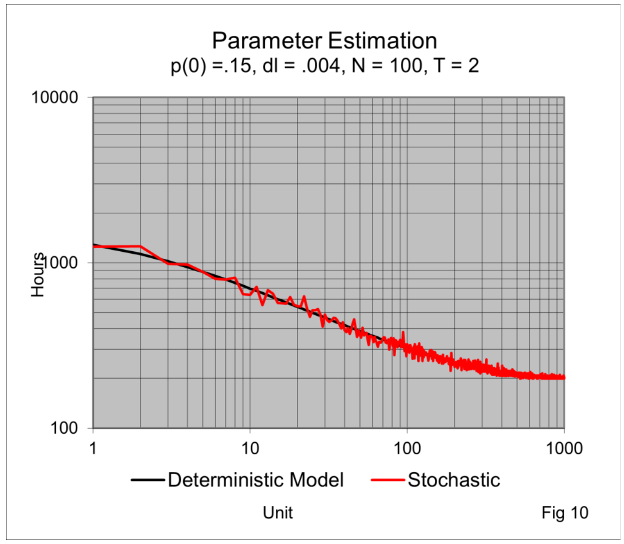
The model calculates labor hours to complete a task by iterating **p(t+1) = p(t) + (1 - p(t)) dl** until x ~ U(0,1) is less than **p(t)**. The average number of iterations for x < **p(t) is 1/p(t).** Thus deterministically,

p(unit+1) ~ p(unit) + (1 /p(t)) (1 - p(t)) dl, where **p(unit)** is the average probability during a unit's trials. The hours(**unit**) are equal to **T N / p (unit).**

Figure 1 shows the calculation. For example for unit 2: **p(unit 2) =** .178 = .156 + 6.4 * (1 - .156) * .004.

Hours(**unit 2**) = 1126 = 2 * 100 / .178.

Table 1		**Deterministic fit**	**Stochastic model**
	p	0.156	0.150
	dl	0.004	0.004
	T	2.06	2.00
	N	97.2	100
Unit	**p(unit)**	**Deterministic Model**	**Iterations**
1	0.156	1284	6.4
2	0.178	1128	5.6
3	0.196	1021	5.1
4	0.213	942	4.7
5	0.227	881	4.4



The parameters of Table 1 generate the learning curve in figure 10. Both the deterministic and stochastic versions are shown.

The parameters of the model are **N**, **T**, **p,** and **dl**. **N** and **T** cannot be separately estimated from just the hours(**unit**) data. However formula 3, **σ/u = ((1-p) / N)**$^{.5}$, provides a means of calculating **N.** **N = (1-p) / (σ/u)**^{2} . It requires the calculation of **σ/u**, the relative standard deviation. One method is to calculate the sequential variation around the expected value of the hours per unit. Then correct the resulting standard deviation for the additional variation introduced by the sequential differencing. That correction factor is the square root of .5. An approximation of the relative standard deviation is thus: **σ /u =** .707 STDev(2 (hours**(nunit+1**) **-** hours**(unit**)) / (hours **(nunit+1**) **+**hours**(unit**)) taken over the range of units.

Once **N** is estimated, Excel's Solver can be used to find the **p(0)**, **dl**, and **T** that minimize the difference between the actual and deterministically modeled learning curve.

Of course, if the actual learning curve has dynamics beyond those modeled by the negative binomial distribution, those dynamics should be modeled. The modeling of design change feedback into manufacturing was introduced

Table 2						
Tasks						
WBS/CE	**Structure**	**Subsystems**	**Avionics**	**Systems Eng**	**Test**	**Total**
Fabrication	125	23	15	8	10	181
Minor Assy	100	69	45	15	20	249
Major Assy	25	138	90	52	70	375
Total	250	230	150	75	100	805

earlier. It can be included in the Table 1 formulation by resetting the **p(unit)** values to **p (0),** in proportion to the ratio of design changes to the total designs at each unit.

Time Domain Formulation

During the design of a product, there may be information flows between engineering tasks. But unlike a production line, these new requirements are applied to the design tasks themselves rather than to a subsequent design. Modeling the equation, **p(t+1) = p(t) + (1 - p(t)) dl,** in the time domain for both engineering and manufacturing allows these information flows to interact within and between the design tasks and factory tasks.

To show these effects a small airplane program is modeled. It produces 300 aircraft over 7 years. Its design effort begins 3 years before the first factory complete aircraft. Table 2 shows 805 engineering tasks in this simulation each with an associated task set in manufacturing.

The engineering tasks are organized into a Work Breakdown Structure of 5 elements, and the corresponding manufacturing tasks into three cost elements. For example, there are 250 Structure design tasks. One hundred twenty-five of those designs are built in fabrication.

With modeling in time, design problems encounter in minor assembly and major assembly can be fed back to engineering where the design is changed and then sent on to the manufacturing elements. This creates a sustaining engineering effort as well as an increase in fabrication and minor assembly hours. In table 3 shows there is a .0006 probability that a trial in a major assembly or minor assembly task will cause a design/fabrication task to be redone, with an additional .0006 that a major assembly trial will cause a design/minor assembly task to be redone. This generates design changes proportional to the hours worked in manufacturing. After the design change is completed it is sent on to its corresponding manufacturing element where its **p** is set to **p(0)**.

Table 3									
Feedback Parameters									
WBS	**Structure**	**Subsystems**	**Avionics**	**Systems Eng**	**Test**	**Cost**			
Structure	0.05	0.05	0.05	0.05	0.05	**Element**	**Fabrication**	**Minor Assy**	**Major Assy**
Subsystems	0.05	0.05	0.10	0.15	0.20	**Fabrication**	**0**	0.0006	0.0006
Avionics	0.05	0.05	0.10	0.15	0.20	**Minor Assy**		**0**	0.0006
Systems Eng	0.05	0.05	0.10	0.15	0.20	**Major Assy**			**0**
Test	0.05	0.05	0.10	0.15	0.20				

Table 4									
Program Parameters									
WBS	**Structure**	**Subsystems**	**Avionics**	**Systems Eng**	**Test**	**Cost Element**	**Fabrication**	**Minor Assy**	**Major Assy**
Start	10/01/23	10/01/23	10/01/23	06/01/23	12/01/24	#1 flow months	21	21	21
Planned complete	12/01/25	12/01/25	12/01/25	06/01/25	09/01/26	Flow red. slope	0.94	0.94	0.94
p(0)	0.003	0.003	0.003	0.003	0.003	p(0)	0.003	0.003	0.003
dl	0.00001	0.00001	0.00001	0.00001	0.00001	dl	0.00001	0.00001	0.00001
Hours/ task	3	3	3	3	3	Hours/task	0.50	0.75	1.25
Tasks	250	230	150	75	100	Tasks	250	272	283

Also, within engineering, .05 of Subsystems completed tasks produce a design change in Structure, while .15 of Systems Engineering produce design changes in Avionics, and .20 of Test produce design changes in Test. These changes are treated like repetitive units in manufacturing, that is, the probability, **p**, is iterated from its last successful completion.

To complete the description of the airplane program table 4 shows the engineering and manufacturing initiating parameters. Figure 11 has the aircraft production schedule. Its blue line shows the beginning of the fabrication effort for each unit. The time domain model is programmed to expend labor hours as needed to complete each unit on time. Thus, the planned complete and complete lines are nearly coincidental. This is of course unrealistic; but was done to keep this paper focused on the learning curve.

Figures 12, 13, and 14 show four simulations, black, red, yellow, and blue, each with an increased level of feedback.

The black lines show the engineering and manufacturing headcounts and learning curve when there is no feedback. The engineering effort is completed on 6/1/26 on the planned schedule. The learning curve slope is 71.6%.

The red lines have feedback initiated only in manufacturing. This feedback produces design changes in engineering, generating sustaining engineering. These design changes are fed back

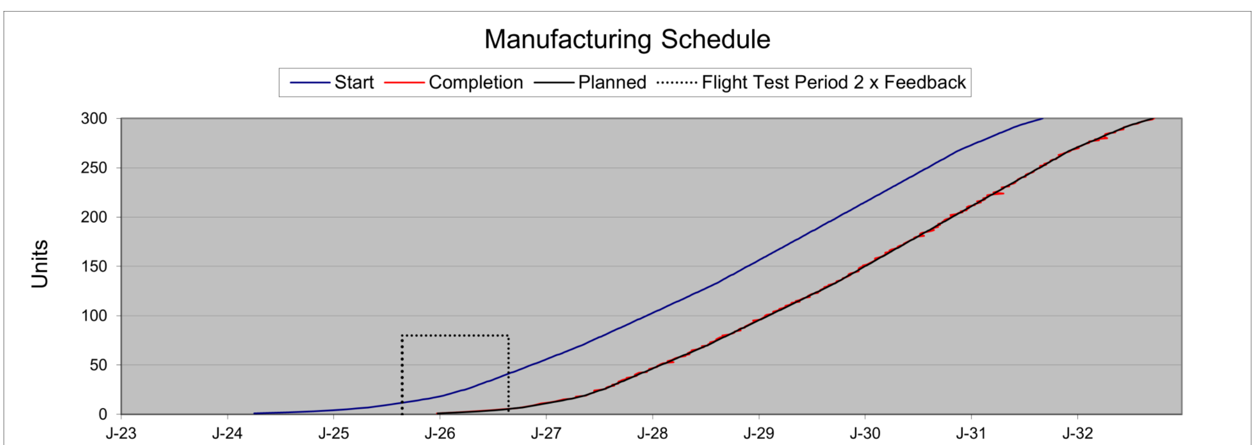


Fig. 11

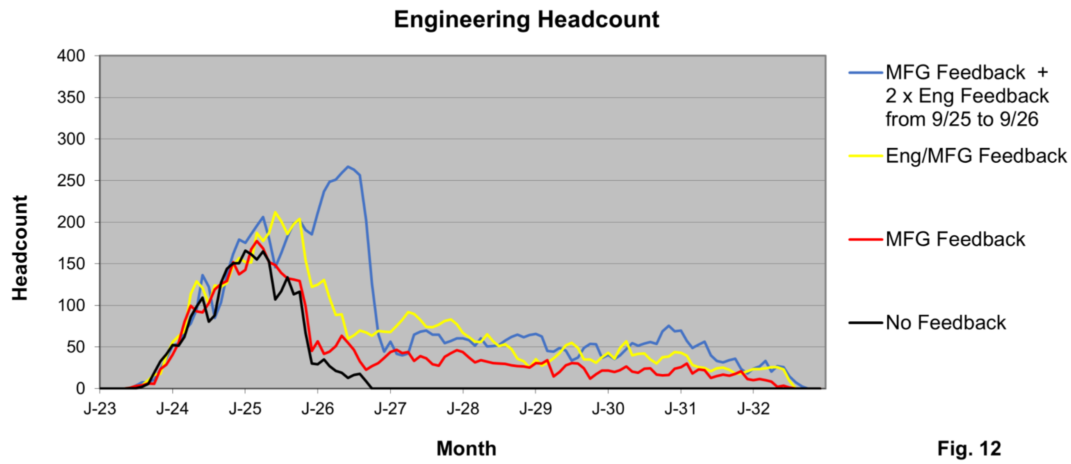


Fig. 12

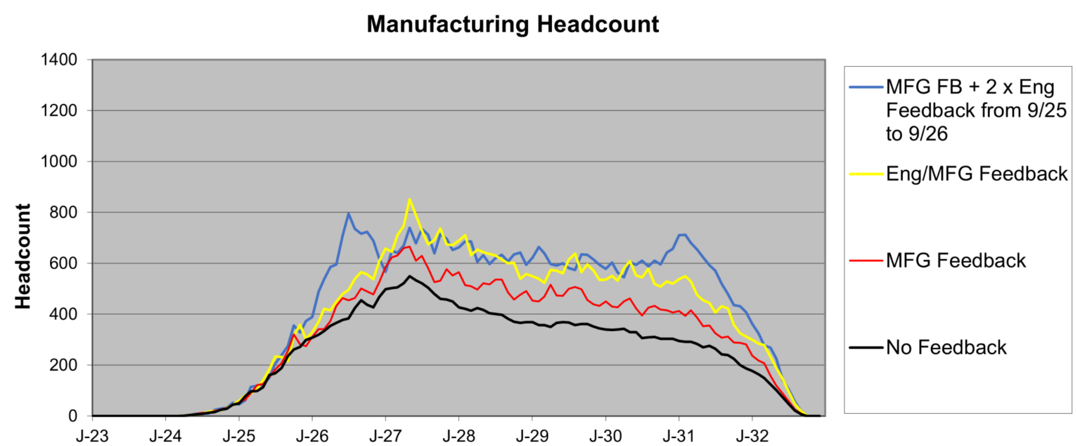


Fig 13

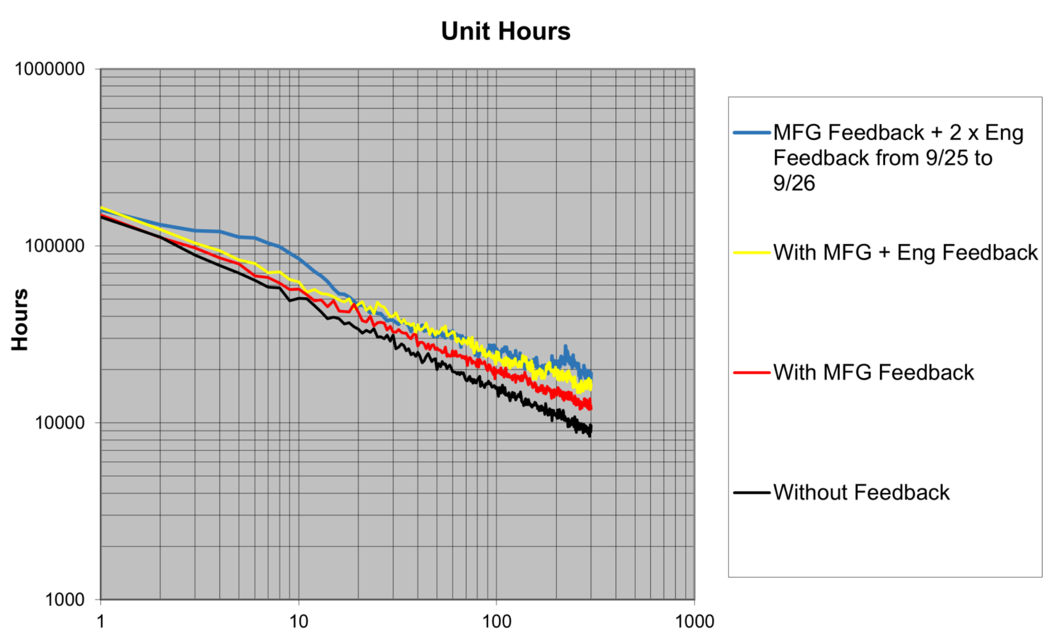


Fig 14

into manufacturing resetting its task's current **p(t)** values to their **p(0)**. The increased resources are seen in the red line of the manufacturing headcount chart and in the flattening of the learning curve.

The yellow lines show the effect of including the internal engineering feedback.

In a project that stretches the engineering capability, due to a new technology, inexperience, or other causes, the design process may have errors. These errors can be expected to show up in the later stages of the first aircraft's production and during the flight test period.

The blue lines illustrate the effect of those errors being doubled between 4 months prior to the first aircraft's completion and the completion of the engineering test effort, depicted as the box in figure 11. The simulated learning curve has the hump often seen in these circumstances. It is followed by a steep decline and then a transition to a more typical curve.

Uncertainty

When this stochastic process is implemented as a Monte Carlo system the predictions are in the form of Probability Density Functions, PDFs. In figures 12 - 14 there are two fundamental causes of variation, the uncertainty of the trial outcomes and the feedback processes.

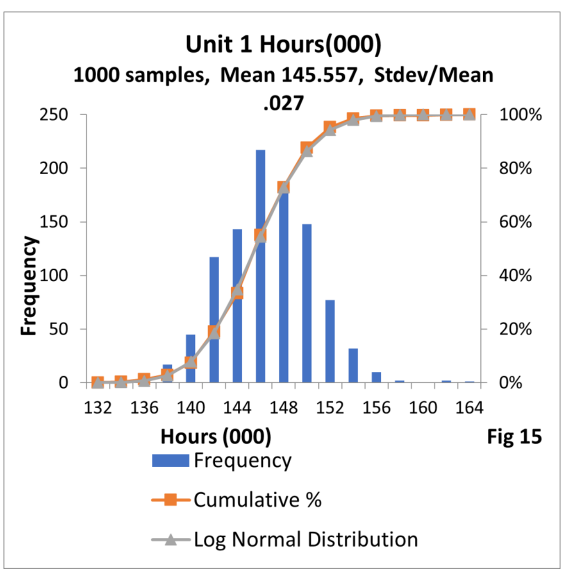


Fig 15

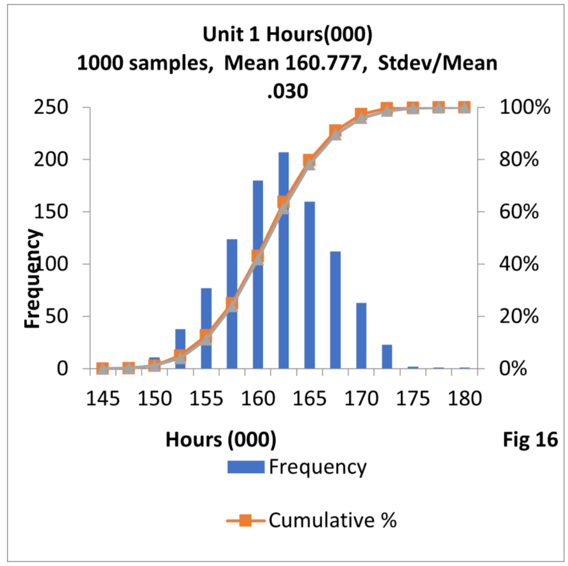


Fig 16

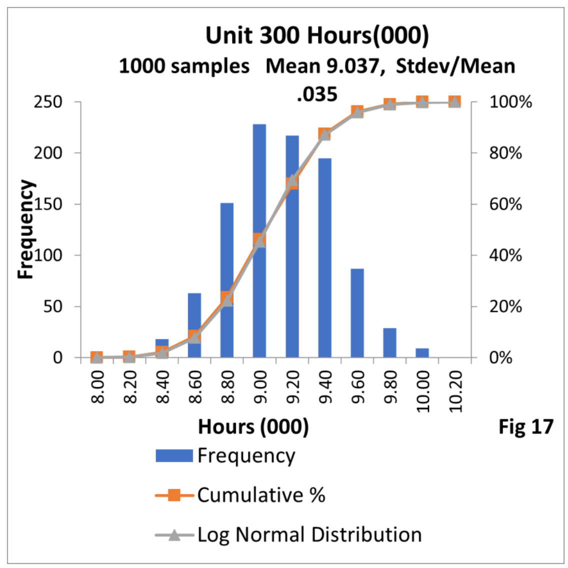


Fig 17

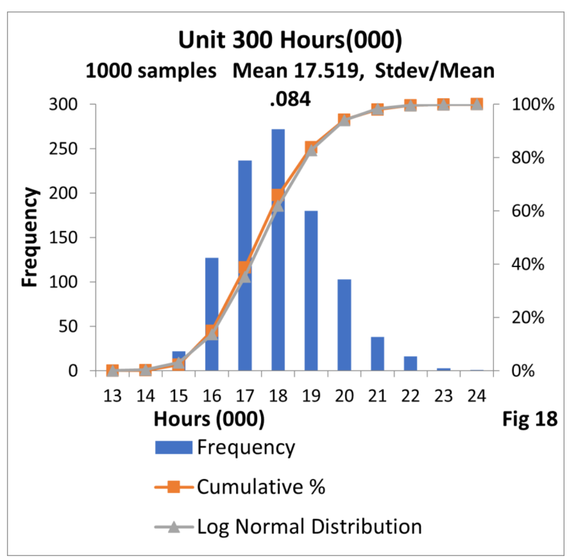


Fig 18

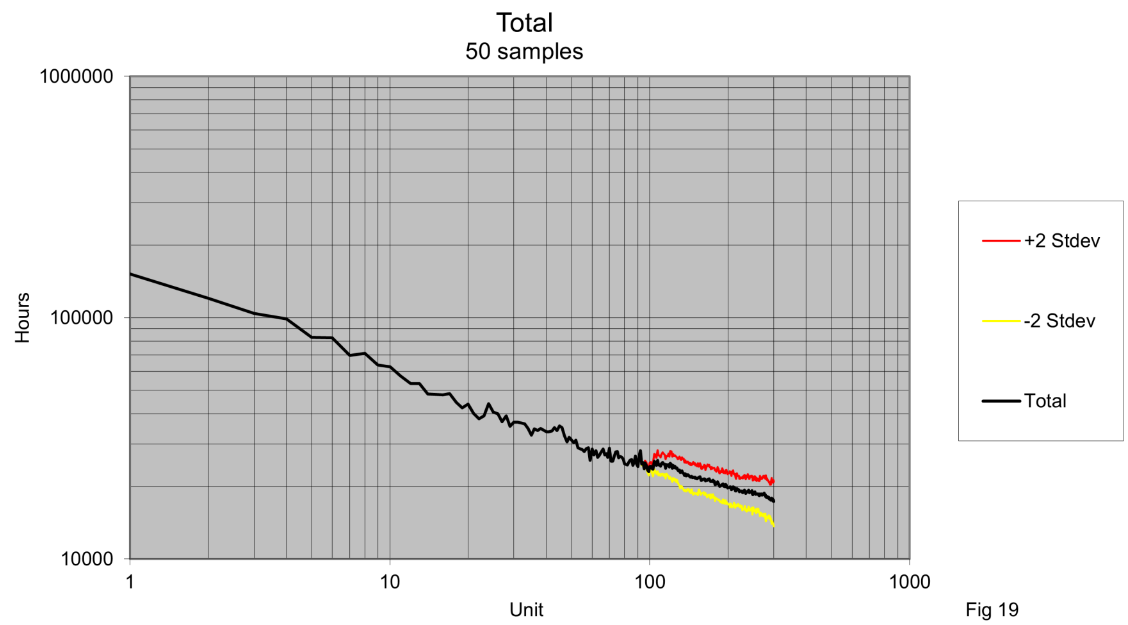


Fig 19

The histograms in figures 15 – 18 are from 1000 iterations of the time domain model. Figures 15 and 17 show the PDFs of unit 1 and 300 hours with only the trial outcome uncertainty, **p(t)** > **x** ~U(0,1). Figures 16 and 18 show the PDFs including the feedback processes; while holding all other input parameters constant. Feedback from design changes increases the variation as the project progresses. Figure 18 shows about twice the variation and hours as the corresponding simulation without design changes, figure 17.

All the distributions are close to lognormal.

Fixing the Monte Carlo model's random number generator to a single sequence for the first 6 years of the simulation produces learning curves with its uncertainty beginning at unit 96, shown in Fig 19, with expanded scales in figure 20.

It is notable that the uncertainty does not grow as one might expect from a typical random walk model. When the iterations of **p(t)**, and thus labor hours, to a successful completion are greater than 1/ **p(t)**, **p(t)** becomes larger than expected. For

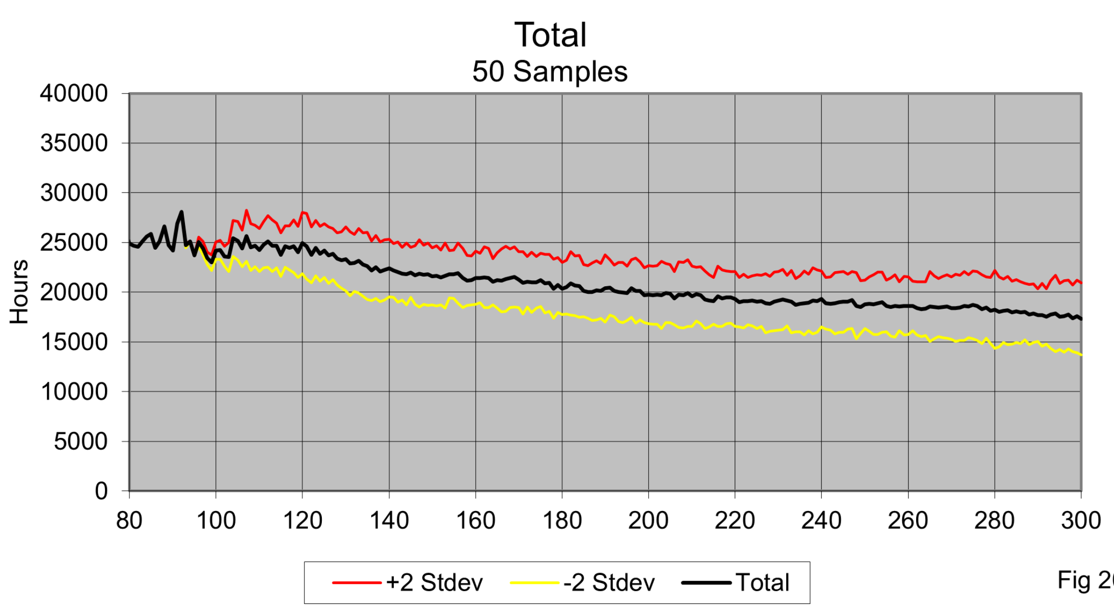


Fig 20

the next unit's iterations that larger probability reduces the expected number of iterations. Thus, a slight autoregressive dynamic is produced.

Information

The model links work to information. Figure 22 shows the bits per successfully completed unit for the two learning curves in figure 21. Information is calculated as:

$$\mathrm{I}(\mathbf{unit}) = (-\sum_{i=t}^{t+n-2} \log_2(1\text{-}\mathbf{p}(\mathrm{i}))) - \log_2(\mathbf{p}(t+n-1)),$$

where t is the first trial of **unit** and ***n*** is the number of trials to the successful completion of **unit.** Both curves have 1000 tasks and a trial time of 1 hour. The black curve, with p(0) = .01 and dl = .00017, has a slope of .717. The blue curve, with a **p(0)** = .15 and **dl** = .003 was chosen to illustrate a curve with a hump that flattens as **p** (large) approaches one.

From an information perspective, work can be thought of as the effort required to resolve an uncertainty. The relationship between a product's complexity, that is uncertainty to be resolved, and the work to resolve it is a subject for information theory. Connecting work with information allows the mathematics of information theory to be brought to bear on the nature of work.

Summary

The iteration of **p(t+1) = p(t) + (1 - p(t)) dl** describes much of the dynamics of the learning curve and work in general. With **N**, the number of tasks, and **p**, a measure of difficulty, both the size and complexity of a project can be modeled. While these parameters can be derived from actual learning curves. By embedding the iteration of **p(t)** into a feedback system the impact of the broader work system can also be evaluated.

The model **p(t+1) = p(t) + (1 - p(t)) dl** states that, given a measure of knowledge **p(t)**, with a trial that knowledge will likely be increased in proportion to the remaining unknown. While the constant of proportionality is on the order of **p (0)** squared. This process creates the learning curve.

It also explains:

Why initial development curves are flat. In some circumstances, the interaction of engineering and manufacturing can create a loss of control in the factory. Prediction is necessary for effective control.

Why learning curves eventually flatten. Knowledge of the process is fully gained.

How processes with a range of process knowledge can have a flatter slope. The sum of a

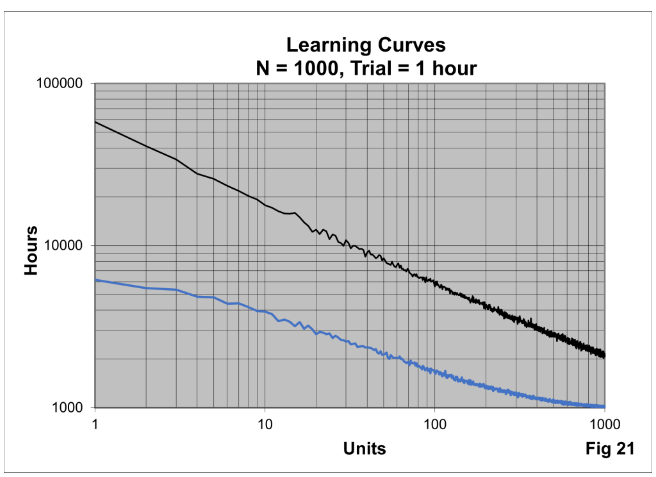


Fig 21

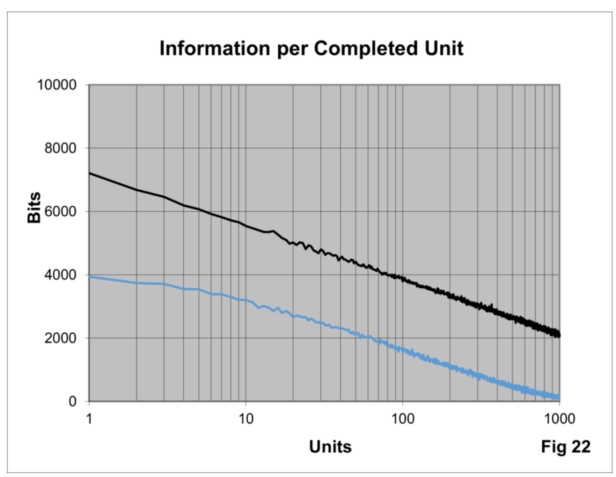


Fig 22

range of slopes creates a learning curve with a flatter slope.

The impact of design changes. The probability of success returns to **p(0)**.

Why assembly slopes are generally steeper than fabrication. Feedback introduces a stream of design change into the earlier stages of manufacturing.

Estimates ultimately are intended to support a decision. Some decision criteria are a linear consequence of the cost estimate. For those, the expected value provides sufficient information. Others are options, a split of the estimates PDF, and the estimate's PDF is required. This model fulfills that requirement.

Harry Larsen *attended the University of Washington, graduating with a mathematics degree in 1966. After graduation, he was hired by the Boeing Company as a production estimator. He began building planning systems for Boeing operations on the company's new supercomputer. Many systems and twenty years later he became the Business Planning Manager of Boeing Marine Systems, a division of the Boeing Company, reporting to the division's general manager. He retired in 2002.*

A Continuance of Marginal Cost Methodology in Project Change Management

Daryl Ono, PhD

Abstract: Change management is an inevitable part in the engineering management of engineering projects so effective change management is critical to determine if the proposed changes add economic value to the project. The marginal cost methodology is proposed to effectively manage change and to parse the changes only to those which add economic value. The marginal cost methodology is valuable in engineering decision making and also facilitates statistical analysis in trade studies for applications to future projects.

Keywords: project change management, engineering economics, life cycle costing, marginal costs

Introduction

Engineering changes ("changes") to project scope are inevitable: the more complex a project is and the longer the project lasts, the more changes can be expected. Changes are defined as adjustments to the original plan and could include additions, deletions, substitutions, repurposing, amongst others. Engineering changes can easily number in the hundreds or the thousands for large projects so effective change management is crucial for the project to stay on schedule and to minimize cost overruns.

This paper will focus on the engineering life cycle cost of initial construction only. It will not cover the project once the equipment, building or facility is put into operations and the associated costs from that point on. Again the focus is only on the engineering project costs before the project begins normal operations.

Often changes are handled on an ad hoc basis and change management tracking systems aren't as robust as necessary to handle complex engineering changes. However, a more important question needs to be addressed – do the proposed engineering changes add value to the project? Large changes will warrant further engineering economic analysis but smaller changes should be tracked and evaluated also especially if there are numerous smaller changes implemented. It is imperative then to carefully track all relevant changes to determine if additional engineering economic analysis should be performed.

Outstanding project management software such as "PROJECTMANAGER", "Microsoft Project", and "Easy Projects" can skillfully manage project changes and provide a repository for all projects, both past and present. Engineering project cost changes can be stored and easily accessed, so this advantage enables effective use of the engineering project cost changes for better engineering decision making.

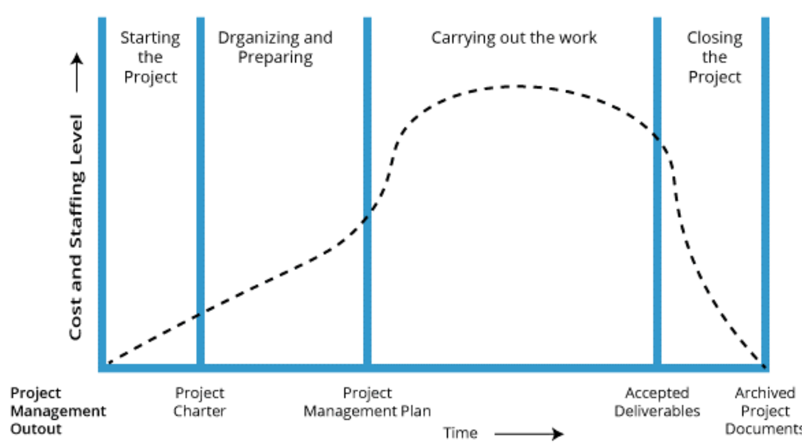


Figure 1.Project Summary

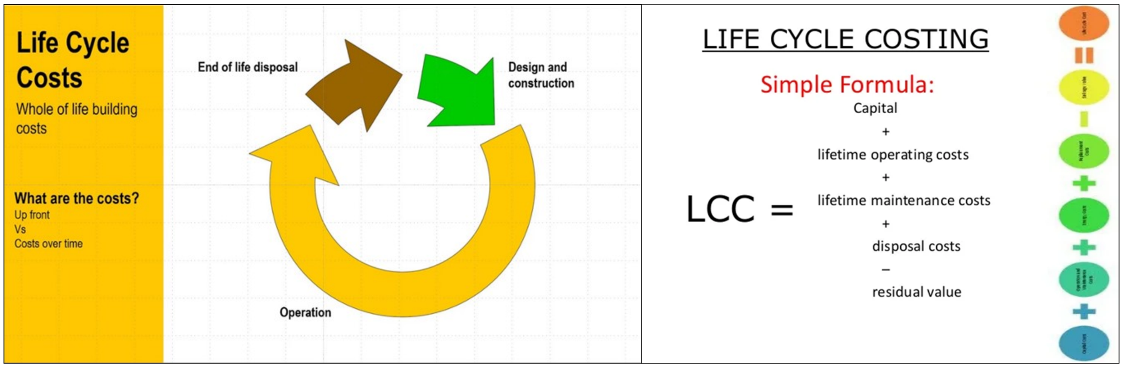


Figure 2.Overview of Life Cycle Costing

Life Cycle Cost Analysis

Life cycle cost analysis studies all costs throughout the lifetime of a project, from inception (including research & development) through the completion of the project. The life cycle cost analysis is parsed into three arbitrary sections: (1) project inception, (2) project operations and (3) project completion. This analysis is done before the project is started and is part of economic feasibility study of the project.

The primary difficulty with life cycle cost analysis is the uncertainty of future costs and the secondary difficulty is future technology. Life cycle cost analysis is time intensive and it's helpful to have experience with these studies (which obviously takes time acquire). The longer the life of the project, the more uncertain the costs are. Finally, technological advances are difficult to anticipate and forecast but could be critical to the success of a project. Project risk will greatly increase if the project involves evolving or cutting-edge technology at any phase.

The effectiveness of life cycle cost analysis is based on the accuracy of the cost inputs into the analysis. Generally acquisition costs are the most accurate forecast but these are only the "tip of the iceberg" and many other costs must be carefully analyzed and incorporated into the life cycle cost analysis. Each project has its own particular costs that are unique to the project and it is dangerous to omit any important cost.

Trade studies of previous projects can mitigate the error of omitting critical costs. Trade studies are a lot of work and it's difficult to see immediate benefits, but the project manager with foresight understands that trade studies are a best practice. Trade studies can be a guide to identify all relevant costs based upon previous projects.

Figure 3. Difficulties of Life Cycle Costing

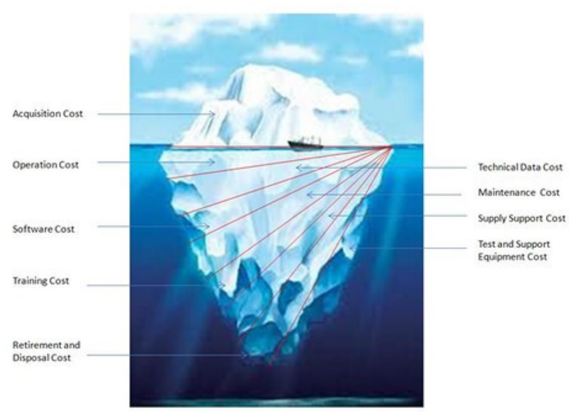


Figure 4. All Relevant Project Costs

Managing Project Change

Once the project has commenced, cash outflows associated with the costs are now expended. As the project progresses these actual costs are accumulated and are compared to committed or budgeted project costs as a cost control mechanism.

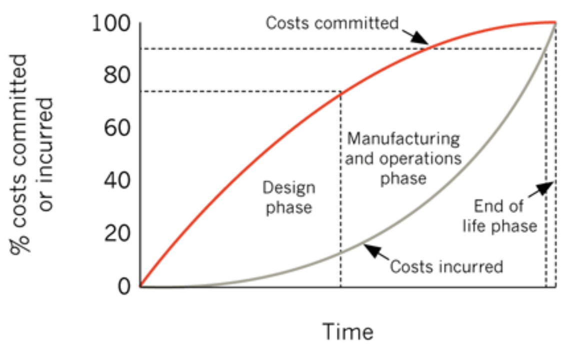


Figure 5.Comparison of Actual Costs to Committed Costs

The project manager and project team gain experience and knowledge as the project progresses so they may propose or institute changes that improve the probability of successful project completion. There could be many proposed improvements and the key is to identify those changes that add economic value to the project. Some proposed improvements could involve increased reliability, cost cutting, process efficiency, etc. Successful project change management includes the proficient administration of these proposed changes. The analysis of the proposed changes could be time-consuming and intense but the larger the proposed changes and its corresponding benefits, the greater the need for this analysis. Finally, as the project progresses, changes will be more difficult to implement, even if the changes are warranted.

The graph above again compares committed costs to actual costs but adds the ease to implement project changes (yellow line). It is important to note the inverse relationship between costs incurred versus ease of change – over time, as project costs accumulate, the ability

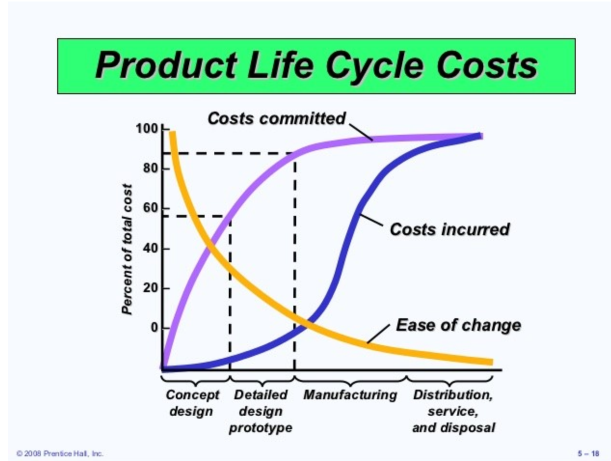


Figure 6. Ease of Implementing Project Changes

to implement changes becomes more difficult. Project changes are relatively easy to execute early in the project's life but becomes more difficult to apply as the project progresses. Finally, the larger the cost of the proposed changes, the greater the need for engineering economics to determine the economic benefits of proposed change to the project.

Overview of Methodology

The marginal cost methodology from microeconomic theory can be applied to the engineering economics of engineering project cost changes. The benefit of the marginal cost methodology is that it can track engineering project cost changes and it can determine the incremental benefits of these changes.

Is the engineering project cost change a change to an existing cost or is it a new additional cost altogether? Is the engineering project cost change a substantial betterment? Does the engineering project cost change alter the project scope? These

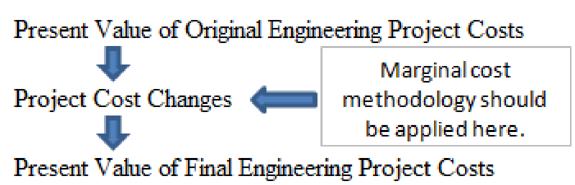


Figure 7. Schematic of Marginal Cost Methodology

are the types of issues that the marginal cost methodology can address in regards to engineering project cost changes. In almost every case the engineering project cost change needs to add economic value for the engineering project cost change to be warranted.

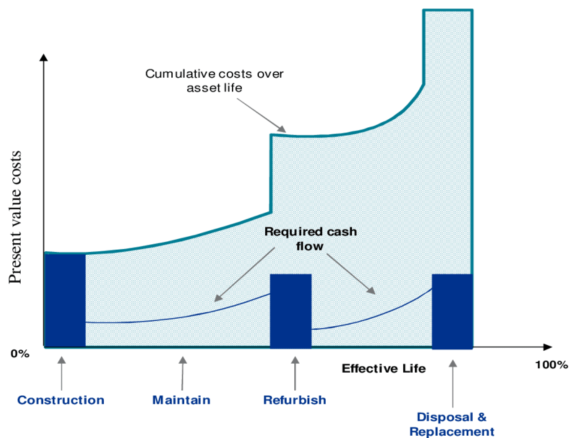


Figure 8. Present Value of Project Costs

Any engineering project cost change that can be eliminated is beneficial especially if this elimination in no way hurts the project scope. It should be investigated why this cost was included in the original analysis but if an engineering project cost can be legitimately eliminated it can only be an advantage to the project.

Time must be a major consideration when estimating the costs of a project. Some US Navy super aircraft carriers take four or more years to construct. Here in the Los Angeles area freeway and highway widening projects progress slowly for around a decade with no completion in sight for the foreseeable future. An article in the *Business Insider* describes how a project to update the US Air Force C-5M Super Galaxy cargo planes took 17 years to complete.

Because of the long timeframes involved time becomes critical in the time value of money component of the engineering economic analysis. Engineering economic analysis must employ the present value methodology to accurately measure the economic value added of an

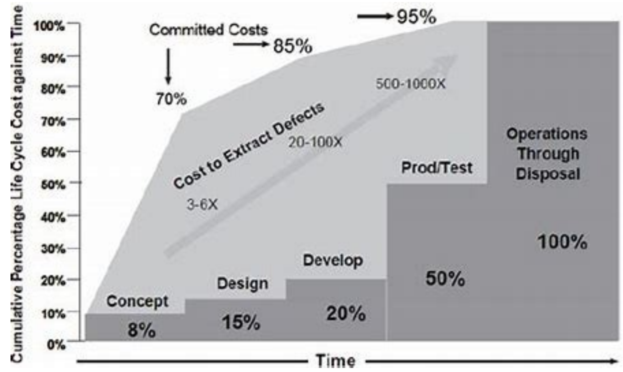


Figure 9. Time as a Critical Component in Engineering Economics

engineering project cost change. Not only is the dollar amount of the engineering project cost change important but the timing of the engineering project cost change is also crucial.

For the purposes of this paper engineering project cost changes will be categorized into three general groupings but there are no conventions that say that these must be the categories. The categories are created based on their timing during their construction cycle and include (1) investment costs, (2) operating costs and (3) terminal costs. Investment costs include engineering design and planning, procurement costs of materials and equipment, licensing and permitting costs, feasibility studies, etc. Operating costs could include construction labor, ongoing procurement costs, supervision, engineering and construction overhead, etc. Terminal costs include disposal costs, inspections, testing, cost of removal, reliability and maintainability estimates, trade studies and documentation, etc. This list is by no means comprehensive and there are many relevant costs and expenses that have been unintentionally omitted.

Committed costs are those cash flows budgeted to the project. Both Figure 5 and Figure 6 show the general pattern of committed costs. Committed costs are low during the project planning stage but hit the maximum once there is the decision to undertake the project. Variance analysis should be conducted between committed costs versus actual costs throughout the project.

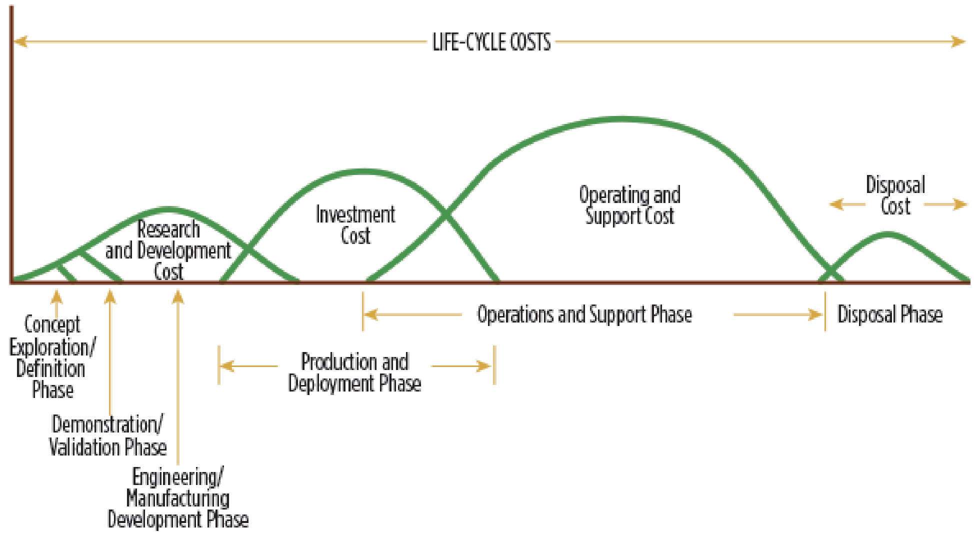


Figure 10. Summary of Life Cycle Costs

Cost control measures should be employed during the project. Ideally actual costs of the project will be at or below committed costs at project completion – the project is over budget if actual costs are greater than committed costs.

Methodology Explained

The general cost function is:

engineering project cost = f(a1, a2, a3)

where a1 is the investment costs, a2 is the operating costs and a3 is terminal costs. Engineering project cost is abbreviated as EPC.

The marginal costs are the following differential equations:

$\partial EPC \div \partial a1 \qquad \partial EPC \div \partial a2 \qquad \partial EPC \div \partial a3$

The objective is to minimize the engineering project cost so it is imperative to understand the underlying structure of the individual costs that constitute the total engineering project cost.

The equation for the engineering project costs is:

$$\text{Present Value of Original Engineering Project Costs} = \sum_{t=0}^{n} \frac{\text{Investment Costs}}{(1+i)^t} + \sum_{t=0}^{n} \frac{\text{Operating Costs}}{(1+i)^t} + \sum_{t=0}^{n} \frac{\text{Terminal Costs}}{(1+i)^t}$$

where i is the hurdle rate used in engineering economics.

The original engineering economics of the engineering project costs contains valuable information imperative for the current project but also to future engineering economics and trade studies. It would be wasteful to discard this information especially as this information can add insight to the final analysis and future trade studies.

The equation for the marginal cost of engineering project cost changes is:

$$\text{Marginal Cost of Engineering Project Cost Changes} = \sum_{t=0}^{n} \frac{\text{engineering project cost changes}}{(1+i)^t}$$

where i is the hurdle rate used in engineering economics and engineering project cost changes are segregated into investment costs, operating costs and terminal costs.

If the engineering project cost change is warranted, feasible and adds economic value, the marginal cost of the engineering project cost changes should be added to the present value of original engineering project costs:

$$\text{Present Value of Final Engineering Project Costs} = \text{Present Value of Original Engineering Project Costs} + \text{Marginal Cost of Engineering Project Cost Changes}$$

The benefit of this approach is that it is straightforward to track various engineering project cost changes to determine economic value added. Instead of updating the present value of original engineering project costs, keep this intact and add the marginal cost of the engineering project cost change to keep the engineering economics updated and integral in engineering decision making.

This marginal cost methodology will help to discern the trend of engineering project cost changes, the timing of engineering project cost changes and determine the incremental economic value added of the engineering project cost changes.

Financial Example

The present value of original engineering project costs was calculated to be $75 million for construction that will last for 3 years. The hurdle rate is 8%. The first engineering project cost change is an increase in Year 1 material costs of $8 million. Labor rate savings are estimated to be $4 million in Year 2 and $2 million in Year 3.

$$\text{Present Value of Final Engineering Project Costs} = \underset{\text{(in \$millions)}}{\$\ 75} + \frac{8}{(1+8\%)^1} - \frac{4}{(1+8\%)^2} - \frac{2}{(1+8\%)^3}$$
$$= \$\ 77.4\ \text{million}$$

The present value of final engineering project costs is updated to $77.4 million.

The marginal cost of engineering project cost changes is calculated as follows:

$$\text{Marginal Cost of Engineering Project Cost Changes} = \underset{\text{(in \$millions)}}{\frac{8}{(1+8\%)^1}} - \frac{4}{(1+8\%)^2} - \frac{2}{(1+8\%)^3}$$

The marginal cost of engineering project cost changes is $2.4 million.

Again, the present value of original engineering project costs remains intact and the incremental engineering project cost changes are added to this amount. The marginal cost methodology can track the level and timing of engineering project cost changes to determine the economic value added of these changes.

Statistical Analysis and Trade Studies

Statistical analysis of engineering project cost changes should be performed to support trade studies. A particular engineering project cost change could make a large difference in a single project but in the long-run which engineering project cost changes are statistically significant? It would aid in engineering decision making to understand which of the individual engineering project cost changes drive the marginal cost of engineering project cost changes in the long-run.

Descriptive statistics are imperative but inferential statistics, particularly regression analysis, should be the statistical tool of choice. The general equation for the regression equation in matrix notation is:

$$\boldsymbol{\beta = (X'X)^{-1}X'Y}$$

where β is the regression coefficients, Y is the dependent variable and X are the independent variables.

In the marginal cost methodology the present value of final engineering costs would be the dependent variable. In this paper the three main engineering project cost changes are categorized into (1) investment costs, (2) operating costs and (3) terminal costs so these would be the independent variables. If the project management system carefully tracked and correctly categorized the engineering project cost changes then the setup of a multiple regression analysis for trade studies should be straightforward. If the independent variables are independent of each other then the multiple regression analysis should provide useful results.

The multiple regression equation where the present value of final engineering project costs is the dependent variable is:

$$\text{Present Value of Final Engineering Project Costs} = \alpha + \beta_1\text{investment cost} + \beta_2\text{operating costs} + \beta_3\text{terminal costs} + \varepsilon$$

where α is the intercept and ε is the error term.

Appropriate regression tests should include ANOVA, t-statistics, correlation analysis and goodness-of-fit (r^2) diagnostics.

Regression analysis can help to determine which of the independent variables (investment costs, operating costs, terminal costs) are statistically significant to plan for regarding engineering project cost changes for engineering decision making.

Was it a change in the level of proposed output of a project once in operation that warranted such as change? If that is true, regression analysis can be implemented to parse cost changes into their fixed and variable component to ultimately perform cost-volume-profit analysis. Linear regression is particularly useful to supplement cost-volume-profit analysis and the linear regression equation is:

$$\text{Engineering Project Cost Changes} = \alpha + \beta(\Delta \text{ in level of ouput}) + \varepsilon$$

where α is the intercept and ε is the error term.

Appropriate regression tests should include ANOVA, t-statistics, correlation analysis and goodness-of-fit (r^2) diagnostics.

Here α can be interpreted as total fixed costs and β as the variable cost per unit. If these two coefficients are statistically valid then cost-volume-profit analysis can be implemented to provide valuable insight into the effects of changes in the level of proposed output. This could explain why an engineering project cost change was necessary.

Case Study - Statistical Analysis

An aerospace subcontractor in the Los Angeles area (which requested anonymity) implements the marginal cost methodology in project change management and has kept accurate records of the present value of original engineering project costs, the present value of final engineering project costs, marginal cost of engineering project changes and engineering project cost changes, which was further segregated into investment costs, operating costs and terminal costs. Regression analysis was performed where the present value of final engineering project costs is the dependent variable and the independent variables are investment costs, operating costs and terminal costs. The table used in the regression analysis follows (in $thousands):

PV Final Engr	Investment	Operating	Terminal
$ 3,471	$ 260	$ 442	$ 23
2,979	225	459	21
4,195	275	478	20
4,701	235	438	19
3,471	240	444	22
3,960	195	379	21
4,701	235	379	20
4,701	265	379	19
3,311	230	386	24
4,664	235	539	17
4,605	302	483	19

Multiple regression analysis was performed and the regression equation is:

		(t-statistics)
PV Final Engr. =	11672	(5.99)
	+ 6.3Investment	(1.66)
	- 5.98Operating	(-2.62)
	- 319Terminal	(-5.63)

$r^2 = .838$ F-calculated = 12.07

Full results are shown in the appendix.

Regression diagnostics for this dataset are generally good but the investment costs are statistically insignificant at the .05 level of significance. A larger sample size could change this conclusion but as of now investment costs do not add explanatory value to the present value of final engineering project costs. Data for investment costs should not be discarded and should be updated along with future additions to operating costs and terminal costs as it could be statistically significant with a larger sample size.

The statistical analysis yields a key insight: the negative coefficients on some of the project changes, specifically changes to operating costs and terminal costs.

PV Final Engr. = 11672
+ 6.3Investment
- 5.98Operating
- 319Terminal

The subcontractor only instituted changes after careful analysis that included the economic benefits of the changes and the timing of the changes. The negative coefficient implies that changes <u>decreased</u> the overall cost of the project. This conclusion is statistically significant. The subcontractor indicated that the project scope remained the same for all of the projects in the statistical analysis. The project remained the same but project costs decreased due to changes which were carefully implemented. Basically, the subcontractor made the right changes at the right time. The changes had the direct economic benefit of decreasing total project costs because of the utilization of effective project change management.

In the case of changes to operating costs, an increase of $1 decreased the total project cost by approximately $6. This is a favorable benefit/cost ratio and shows the advantages of efficient project change management. The subcontractor indicated that changes were handled on an ad hoc, case-by-case basis because each project was unique, but the subcontractor followed the principles of project change management for each and every change. Unfortunately a “one size fits all” is inapplicable for aerospace subcontracting, but a disciplined approach to project change management can generate substantial benefits for a project. The type of project really doesn’t matter.

Other statistical tests can be performed beyond the regression analysis presented in this case study.

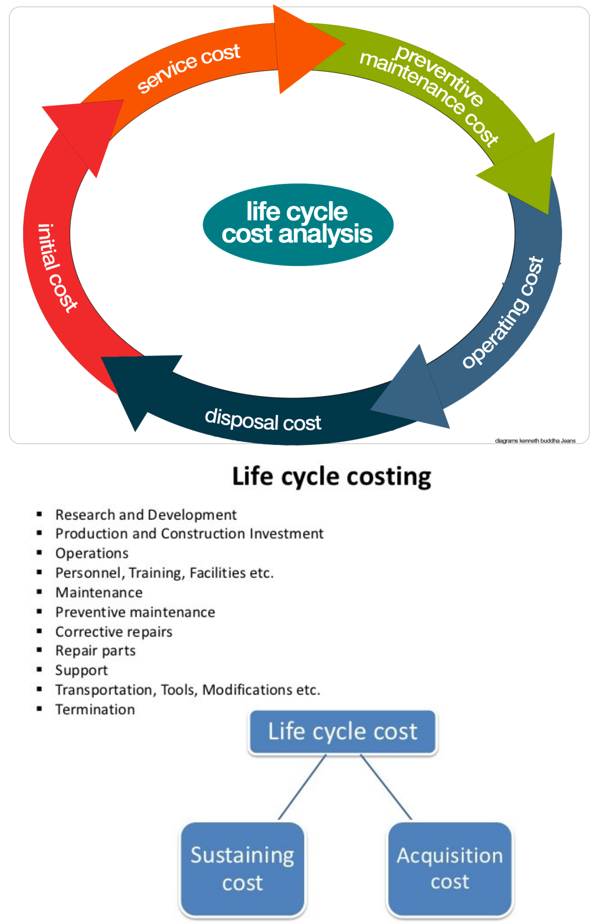
Limitations and Constraints

Obviously, a larger sample size with favorable results would add greater credence to the statistical analysis just presented. The constraint to consider is that the data used in the statistical analysis is not public data and the anonymous subcontractor was kind enough to allow the authors to use its data. Additional studies in this area would therefore depend on other subcontractors to provide data for further analysis. In the hypercompetitive environment of aerospace subcontractors in the Los Angles area, competitors are unlikely to pool data for analysis and cooperate with each other. The aerospace subcontractors generally are unwilling to work with competitors, which make data acquisition difficult at best.

The subcontractor the authors worked with follows the concepts of the Project Management

Institute (PMI) and its project cost accumulation was configured to facilitate statistical analysis. Companies are not required to configure their cost data in this manner. If the authors received data from other subcontractors whose computer architecture does not facilitate statistical analysis, the authors would be required to rearrange the cost data – this could introduce translation errors. Project costs must be configured by the subcontractor's computer architecture, not the authors. The authors could reconfigure the data but that could be an error-prone process.

The authors chose only three project cost categories: (1) initial investment (including R&D), (2) operations and (3) termination. A finer breakdown could have been used that would have introduced additional cost categories, although total project cost would remain unchanged. A finer breakdown of cost categories could have been implemented but again the authors studied the cost categories currently in use by the subcontractor. The subcontractor has over a dozen years of experience and these three cost categories were more than adequate for the subcontractor. Again the authors could have added additional cost categories in an attempt to derive additional intuition of project costs and this is something that the authors must consider. The graphics on the right are suggestions for the further dichotomy of costs.



Conclusion

The marginal cost methodology in project change management is a technique to successfully manage changes to project scope in the construction of projects. This is especially important for very expensive projects and projects which will take an extended period to complete. It facilitates both engineering economics and statistical analysis. The conclusions in the marginal cost methodology provide valuable insight to both current and future large-scale engineering projects but is equally beneficial to small or medium-sized projects that must contend with numerous changes.

Project changes that are subjected to feasibility reviews, that are analyzed by employing disciplined quantitative techniques and that are implemented on a timely basis add economic value to a project.

Appendix: Regression Results

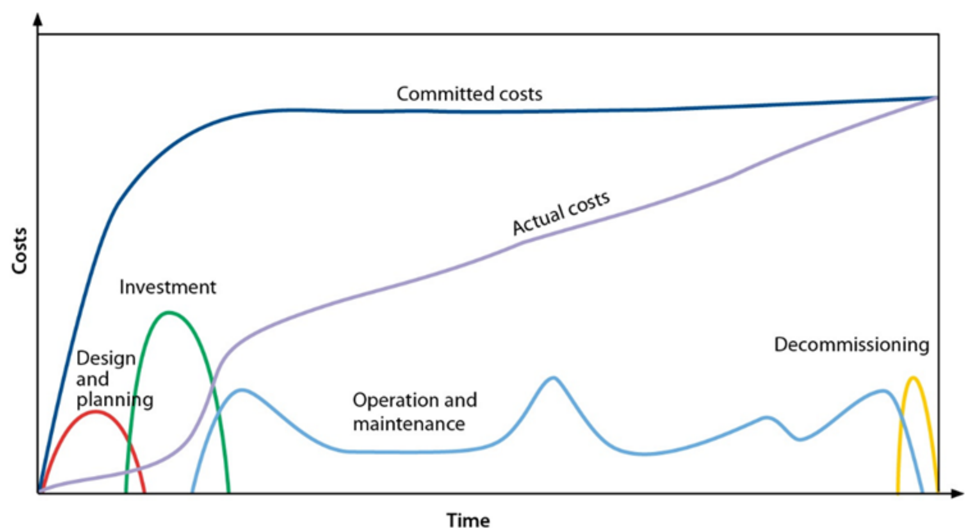
SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.915443256
R Square	0.838036354
Adjusted R Squa	0.768623363
Standard Error	317.0777276
Observations	11

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	3	3641454.003	1213818.001	12.07319179	0.003721247
Residual	7	703767.9973	100538.2853		
Total	10	4345222			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	11672.4257	1949.059081	5.988749041	0.000548381	7063.633333	16281.21807	7063.633333	16281.21807
Investment	6.268066369	3.766390671	1.664210359	0.140017229	-2.63803235	15.17416509	-2.63803235	15.17416509
Operating	-5.980690111	2.284299167	-2.618172872	0.03450074	-11.38219932	-0.579180905	-11.38219932	-0.579180905
Terminal	-319.1087158	56.69490629	-5.628525324	0.000791997	-453.1708661	-185.0465655	-453.1708661	-185.0465655



References:

Agarwal, Prateek (July 25, 2018). Theory of Production – Cost Theory. Retrieved from https://www.intelligenteconomist.com/theory-of-production-cost-theory/

American Society of Civil Engineers (n.d.). *Maximizing the Value of Investments Using Life Cycle Cost Analysis.* Retrieved from http://www.asce.org/uploadedFiles/Issues_and_Advocacy/Our_Initiatives/Infrastructure/Content_Pieces/asce-eno-life-cycle-report.pdf

Beal, V. (n.d.). *SQL - Structured Query Language.* Retrieved from https://www.webopedia.com/TERM/S/SQL.html

Brigham, E. F. and M. C. Ehrhardt. Financial Management. 12th Edition, 2008.

BusinessDictionary (n.d.). *Relational Database.* Retrieved from http://www.businessdictionary.com/definition/relational-database.html

Campus Academic Resource Program (n.d.). *Cost-Revenue-Profit Functions (Using Linear Equations)*. Retrieved from file:///H:/Handouts/CostEquationProblems.pdf

Chand, Smriti (n.d.). *Cost Theory: Introduction, Concepts, Theories and Elasticity.* Retrieved from http://www.yourarticlelibrary.com/economics/cost-theory-introduction-concepts-theories-and-elasticity-economics/28725

Easy Projects *(n.d.)*. Retrieved from https://try.easyprojects.net/engineering?utm_source=bing&utm_medium=cpc&utm_campaign=USA%20-%20Industries&utm_term=engineering%20project%20management%20software&utm_content=Engineering%20PM%20Software

Fleming, C. C. and B. von Halle. *Handbook of Relational Database Design. 1989.*

Hillier, F. S. and G. J. Lieberman. *Introduction to Operations Research.* 7th Edition, 2001.

Hubbard, R. G. and A. P. O'Brien. *Microeconomics.* 3rd Edition, 2010.

Johnston, J. *Econometric Methods.* 2nd Edition, 1972.

Maddala, G. S. *Econometrics.* 1977.

Labdesignnews.com (n.d.). *Distribution of Costs by Trade.* Retrieved from https://www.bing.com/images/search?view=detailV2&ccid=0H0q%2b5Up&id=04A7B831B553BD62A45A7CFFA5D16C1089B59A4D&thid=OIP.0H0q-5Uph38aXmUd9E9OLAHaHa&mediaurl=http%3a%2f%2fwww.labdesignnews.com%2fsites%2flabdesignnews.com%2ffiles%2flegacyimages%2fRD%2fLab_Design_News%2fArticles%2f2009%2f07%2fld976_lead_fig1pie%25282%2529.gif%3fn%3d3659&exph=400&expw=400&q=construction+cost+images&simid=607987494117966488&selectedIndex=71&ajaxhist=0

Life Cycle Engineering (n.d.). *Life Cycle Cost Analysis.* Retrieved from https://www.lce.com/Life-Cycle-Cost-Analysis-238.html

Maier, D. *Theory of Relational Databases.* 1983.

Nicholson, W. *Microeconomic Theory*, 3rd Edition, 1985.

PROJECTMANAGER (n.d.). Retrieved from https://www.projectmanager.com/industries/engineering-project-management?utm_source=bing&utm_medium=cpc&utm_campaign=ENGI-Industry-US&utm_term=engineering%20project%20management%20software-exact

Sullivan, M. *Statistics.* 2nd Edition, 2007.

Techopedia (n.d.). *Relational Database (RDB)*. Retrieved from https://www.techopedia.com/definition/1234/relational-database-rdb

University of Calicut (n.d.). Mathematical Economics and Econometrics. Retrieved from http://www.universityofcalicut.info/SDE/BA_economics_mathematical_economics_and_econometry.pdf

Varian, H. R. Microeconomic Analysis. 2nd Edition, 1984.

Winston, W. L. and S. C. Albright. *Practical Management Science.* 5th Edition, 2016.

wiseGEEK (n.d.). *What is a Relational Database?* Retrieved from http://www.wisegeek.org/what-is-a-relational-database.htm

Wolf, Carl, PhD, PE (n.d.). *Engineering Cost Analysis*. Retrieved from https://web.njit.edu/~wolf/intro.htm

Woody, Christopher (August 20, 2018). *After 17 Years of Upgrades, the Air Force's Biggest Plane is Ready to Stay in the Air for Decades*. Retrieved from https://www.msn.com/en-us/money/companies/after-17-years-of-upgrades-the-air-forces-biggest-plane-is-ready-to-stay-in-the-air-for-decades/ar-BBLTkSz?li=BBnb7Kz&ocid=mailsignout

Daryl Ono*, MBA, MS, MS, MS, MSEM, PhD, CIA, CCE/A is an NTT associate professor at Occidental College and an adjunct instructor in the PhD Program in Global Leadership at the Indiana Institute of Technology. He has worked with aerospace engineering firms in the Southern California area for the last two decades. Mr. Ono has an MS in systems engineering from Southern Methodist University, an MBA from the Claremont Graduate School and an MS in operations research from Southern Methodist University.*

The International Cost Estimating and Analysis Association is a 501(c)(6) international non-profit organization dedicated to advancing, encouraging, promoting and enhancing the profession of cost estimating and analysis, through the use of parametrics and other data-driven techniques.

www.iceaaonline.com

Submissions:

Prior to writing or sending your manuscripts to us, please reference the JCAP submission guidelines found at

www.iceaaonline.com/publications/jcap-submission

Kindly send your submissions and/or any correspondence to JCAP.Editor@gmail.com

International Cost Estimating & Analysis Association

4115 Annandale Road, Suite 306 | Annandale, VA 22003

703-642-3090 | iceaa@iceaaonline.org