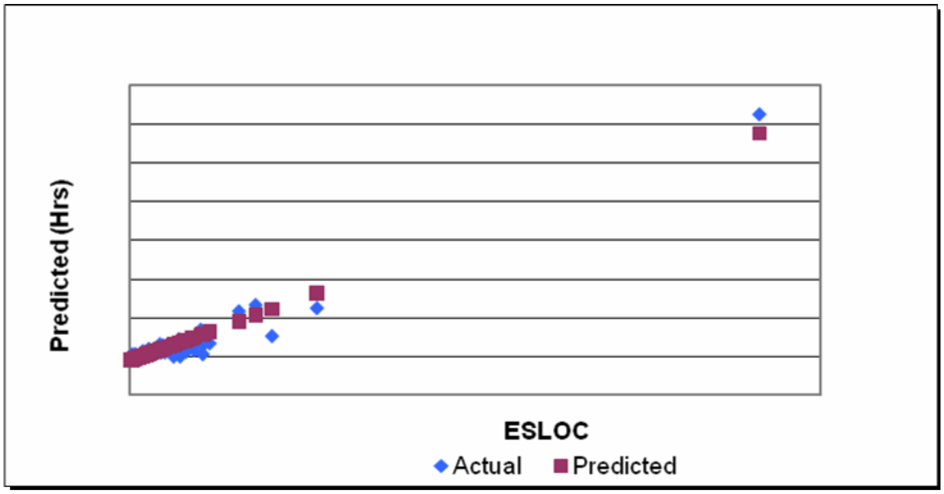
Software Cost Estimating Relationships



Abstract: Software cost overruns are a common problem for the majority of software development projects. With the ever increasing amount of software present in current Department of Defense (DOD) programs, it is extremely important to generate accurate software cost estimates. There are many complex models that estimate software development productivity and costs. This paper builds upon the principles of these models to look for a simple regression model that can be used to generate accurate and defendable cost estimates for software development programs.

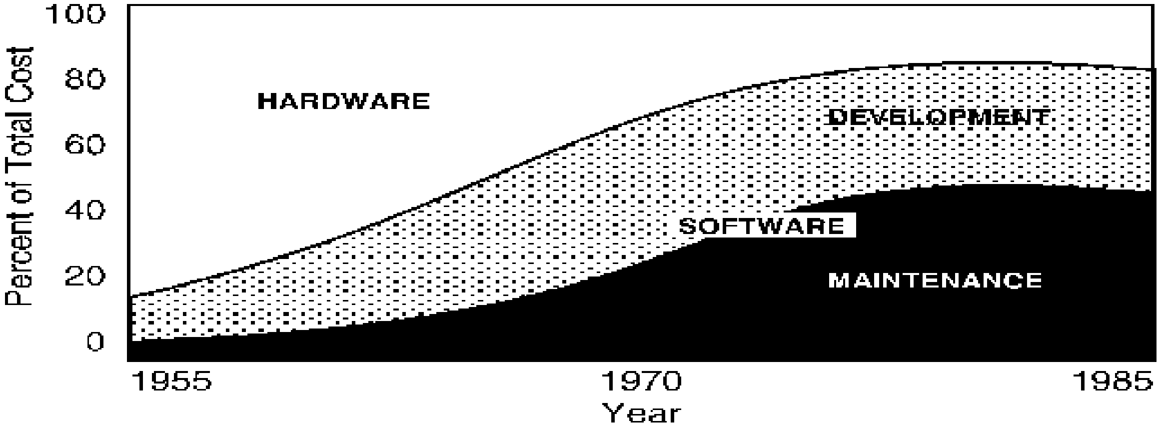
Table of Contents

I. Introduction

The expansion in everyday computer use and computer hardware capabilities has produced increased demands on the capacity of software programs (National Aeronautics and Space Administration, 2002). In fact, the capabilities of current and future military systems are dependent on the performance of a systems' software (National Aeronautics and Space Administration, 2002). As a system is upgraded or improved, much of the additional capability is achieved through new software (National Aeronautics and Space Administration, 2002). Due to software's flexibility, the Department of Defense's appetite for software has been described as "virtually insatiable" (National Aeronautics and Space Administration, 2002). Furthermore, the majority of programs that are procured by the Department of Defense (DOD) include some software. Whether it is the avionics on the Joint Strike Fighter, the software "guts" of a radar, or the programs for a computer based training module; software is an integral part in the development of these systems.

As time goes on, software costs have grown to be much larger than hardware costs, as a percent of total program cost (National Aeronautics and Space Administration, 2002). This changing relationship is illustrated in Figure 1 (National Aeronautics and Space Administration, 2002):

FIGURE 1



While software does provide many opportunities for increasing a program's capability, software development has its pitfalls. One of the main problems with software development is its consistent overrun of estimated cost and schedule targets. Approximately 1/3 of all programs are delivered late **and** exceed their budgets; while 2/3 of all major development programs substantially overrun their original cost estimates (Wu, 1997). In order to prevent these cost overruns, various methods of estimating the cost of software projects have been developed.

II. Software Estimating: Background

The first contribution to estimating software development was in 1958 with the introduction of the Norden staffing profile (Jensen, Putnam, & Roetzheim, 2006). This technique has subsequently been incorporated in many estimating methodologies used today (Jensen, Putnam,

& Roetzheim, 2006). Many of the software estimating tools used today (PRICE S, SEER, COCOMO, etc.),[1] were first introduced in the mid-1970s (Jensen, Putnam, & Roetzheim, 2006). Most of these estimating tools were originally developed during 1974-1981 with occasional changes and updates, such as refining the algorithms and cost drivers (Jensen, Putnam, & Roetzheim, 2006).

The simplest method employed to create software cost estimates is an equation that can be used to calculate the total effort of the program primarily through its size; Equivalent Source Lines of Code (ESLOC) (Jensen, Putnam, & Roetzheim, 2006). This equation is:

$$E_d = C_k S_e \qquad (1)$$

Where:
E_d = the development effort in man hours
C_k = the productivity factor (defined as man-hours/ESLOC) and
S_e = the number of ESLOC

Although this equation remains popular within the cost community, it has evolved into a more complex equation that takes into account not only the software's size and diseconomies of scale; but numerous environmental factors that can affect software development (Jensen, Putnam, & Roetzheim, 2006). These complex models generally use about 25 different factors to adjust for outside factors (Maxwell & Van Wassenhove, 1999). A general representation of this is:

$$E_d = C_k \sum_{i=1}^{n} f_i S_e^{\beta} \qquad (2)$$

Where:

E_d = the development effort in man hours

---

[1] The Constructive Cost Model (COCOMO) is one of the first parametric models developed. It is currently in the public domain and has a number of COCOMO-based variants. There are three versions of COCOMO: Basic-COCOMO, Intermediate-COCOMO, and Detailed-COCOMO. The most commonly used is the intermediate version. Development time is estimated using input parameters in four categories (product, computer, project, and personnel). The original COCOMO has been expanded and refined by its inventor, Barry Boehm, and a team from the University of Southern California. It has been replaced by a version called COCOMO II. COCOMO II, while mathematically similar to the original COCOMO, uses much more sophisticated size estimation techniques and a more complex formula. The Parametric Review of Information for Cost Estimating Software (PRICE-S) is a proprietary commercial parametric model originally developed by Martin-Marietta. It includes a productivity factor that is calibrated to the developer's software engineering environment. Several additional parameters (support schedule, number of installations, expected growth, quality levels, etc.) are used to estimate the support effort. The Software Evaluation and Estimation of Resources - Software Estimating Model (SEER-SEM) is a proprietary commercial model marketed by Galorath Associates. It uses 'knowledge-bases' that are a function of a project's platform, application, and development standards. SEER-SEM also considers the development paradigm being used and estimates support parameters by including anticipated support years, software maintenance changes, number of sites, growth factor, and required support level (Defense Acquisition University, 2008).

C_k = the productivity factor (defined as man-hours/ESLOC)
f_i = the i^{th} environmental factor
n = the number of environmental factors
S_e = the number of ESLOC
β = an entropy factor that accounts for the productivity change as a function of effective product size

While these models can be very effective, they are generally 25 years old (Jensen, Putnam, & Roetzheim, 2006) and the formulas used to generate estimates are not accessible to the public since most of the models are proprietary. This "black box" approach makes it more difficult for analysts to defend their estimates, due to lack of insight into the data. Moreover, even though all of these models exist, software cost overruns remain a persistent problem (National Aeronautics and Space Administration, 2002). The following analysis uses publicly accessible Department of Defense (DOD) databases to look for a simple linear regression that can be used to generate an effective software cost estimate.

III. Data

In order to find a workable and simple cost estimating relationship (CER), the data for completed projects from the Software Requirements Data Reporting (SRDR) database was used (time period ending in October 2008) (Defense Cost Resource Center, 2008). First, an examination of the data was done in order to delete extraneous data points and those projects included in the database that were not completed. A total of 664 data points were left after this examination.

Next, the data was broken up into 20 different data sets (one which included all 664 observations). This was done in order to counter for some of the problems that occurred when running a general regression on the overall dataset. Several different methods of dividing the data were investigated. One method was to divide the data by contractor in order to get a more unbiased look at variables such as productivity and peak staff without outside factors contributing. The data was also divided by commodity since there have been vast differences observed between military, space, and industrial software development applications. Along these lines the data was also divided by military branch to determine if there were stronger relationships for software development for one service over the other. Additionally, data was also divided by several key development features in order to determine if there is any difference between different either programming language or development paradigm. However, data was only divided one time; there was no dataset that sorted the date by more than one category.

The following provides a list of all the ways that the data was broken up as well as a list of the different data sets that were used:

- Overall
- Data Broken out by Development Paradigm
    - Incremental
    - Spiral
    - Waterfall

- Data Broken out by Primary Development Language
    - C (including C, C++, and C#)
    - Java
    - Ada
- Data Broken out by Type of Program:
    - Avionics
    - C4I
    - Mission Planning
    - Training
- Data Broken out by Service:
    - Air Force
    - Army
    - Navy
- Data Broken out by Contractor
    - BAE
    - Boeing
    - General Dynamics
    - Lockheed Martin
    - Northup Grumman
    - Raytheon

In order to ensure that there would be at least 10 degrees of freedom for each of the tested equations, some potential datasets were eliminated. Next, 18 different linear regressions were run on each of the 20 data sets (for a total of 360 regressions run). All regressions were run using CO$TAT 7.1. Some of the data points did not have values for each variable and were automatically omitted from the equation by CO$TAT. The following table shows a summary of the dependent and independent variables:

TABLE 1: ALL VARIABLES USED IN THE EQUATIONS

Independent Variables (Quantitative)	Independent Variables (Dummy)	Dependent Variables
Equivalent Source Lines of Code (ESLOC)	CMMI[2]	Duration (Hours)
Productivity (Hours/ESLOC)		Productivity (Hrs/ESLOC)
Software Requirements		
Peak Staff		

A brief definition of each of these variables is included in the Appendix. The appendix also includes the matrix showing each regressions for each dataset and whether each it was statistically significant or not.

---

[2] The CMMI is divided into five maturity levels (with 1 being the worst and 5 being the best). For the regressions, CMMI was determined to be 1 if the reported level was greater than or equal to 3, and 0 if it was left unrated or less than 3.

The selection of these variables was based on past, observable relationships between the data. In past studies, ESLOC is considered the best way to estimate software costs and is assumed to have a positive relationship with duration and productivity (Ross, 2008). However, this analysis hoped to take into account the effect of team size and teamwork on total development effort. In order to do that, peak staff and the CMMI level were examined.

Several relationships were first examined to determine how the dependent variables respond to the independent variables for all of the datasets (Ross, 2008):

Duration (Hours): Program size (ESLOC)
Productivity (Hours/ESLOC): Peak Staff

It is expected that Duration: ESLOC will result in a positive relationship and Productivity: Peak Staff will result in a negative relationship (Ross, 2008). In other words, the benefits from teamwork display diminishing marginal returns; at some point adding an additional person would actually *increase* the amount of time required to complete a project (Hoegl & Georg Gemuenden, 2001). Therefore, it is expected that as the number of peak staff increases, the amount of productivity will decrease. Building on this often observed fact, the analysis attempted to examine the quality of management and teamwork present for the development programs.

Perhaps one of the hardest elements to measure of a software development program is the quality of the teamwork. A correlation exists between team members' feelings of personal success and the amount of work satisfaction and learning (Hoegl & Georg Gemuenden, 2001). Furthermore, one of the main aspects of teamwork is the relationship between the managers and the employees (Hoegl & Georg Gemuenden, 2001). For example, a manager's lack of detailed information regarding the specifics of a project generally indicates that cost overruns are possible. In order to attempt a measure of this "squishy" quality, the Capability Maturity Model Integrated (CMMI) factor was used to determine the quality of management.

Once these basic relationships were established, different combinations of the variables were examined in order to derive a simple CER using no more than four independent variables to explain either the time spent to develop the software (in hours) or the software developer's productivity (as measured in hours/ESLOC).

IV. Regression Results

After running the regression for each data set, the statistically significant results were examined in more detail. The results show the only equation that consistently shows statistically significance is Duration = ESLOC. As expected, there is a positive relationship between size of the program (measured in ESLOC) and the duration of the effort (measured in hours). The following table shows the predicted and actual relationships for the two relationships discussed previously:

TABLE 2: PREDICTED VS ACTUAL RELATIONSHIPS

Dataset	Duration: ESLOC		Productivity: Peak Staff	
	Predicted	Actual	Predicted	Actual
Overall	+	+	-	-
Incremental	+	+	-	-
Sprial	+	+	-	-
Waterfall	+	+	-	-
Ada	+	+	-	-
C	+	+	-	-
Java	+	+	-	-
Avionics	+	+	-	-
C4I	+	+	-	-
Planning	+	+	-	-
Simulation	+	+	-	-
Training	+	+	-	+
Air Force	+	+	-	-
Army	+	+	-	-
Navy	+	+	-	-
BAE	+	+	-	-
Boeing	+	+	-	+
General Dynamics	+	+	-	-
Lockheed Martin	+	+	-	-
Northup Grumman	+	+	-	+
Raytheon	+	+	-	-

(+) Positive relationship
(-) Negative relationship

From this table we can see that the predicted relationship is very similar to the resulting analysis. The only exceptions are found in the Productivity: Peak Staff relationship; for the Training, Boeing, and Northup Grumman databases. This implies the relationship, while still observable in most cases, is not perfect for all cases.

Even though many equations returned statistically significant results, the R^2 and standard deviations were poor for most of the results. The following tables show the key statistics for these two relationships:

TABLE 3: KEY STATISTICS FOR DURATION: ESLOC

Duration = ESLOC						
Database	Equation	Adjusted R^2	Mean	Standard Error	Coefficient of Variation (Fit Space)	Obs
Overall	Hrs = 6.873e+004 + 0.05726 * ESLOC	-0.02%	72,770.47	199,382.44	273.99%	568
Incremental	Hrs = 3.685e+004 + 0.7119 * ESLOC	46.72%	99,119.06	70,894.61	71.52%	90
Sprial	Hrs = 4.562e+004 + 0.0359 * ESLOC	2.06%	49,695.26	84,481.57	170.00%	140
Waterfall	Hrs = 35898 + 0.3278 * ESLOC	25.44%	58,549.11	83,872.29	143.25%	119
Ada	Hrs = (-34255) + 2.141 * ESLOC	56.85%	120,129.25	201,073.29	167.38%	81
C	Hrs = 2.372e+004 + 0.507 * ESLOC	36.68%	66,193.25	89,178.70	134.72%	291
Java	Hrs = (-50558) + 1.606 * ESLOC	93.72%	104,981.83	82,547.30	78.63%	99
Avionics	Hrs = 5.533e+004 + 0.4753 * ESLOC	17.34%	90,215.18	100,253.11	111.13%	32
C4I	Hrs = 1.296e+004 + 0.5464 * ESLOC	21.37%	42,640.65	52,812.11	123.85%	43
Planning	Hrs = (-1.263e+004) + 1.659 * ESLOC	92.30%	73,784.18	47,010.99	63.71%	29
Simulation	Hrs = 57990 + 0.1603 * ESLOC	3.69%	82,848.86	135,201.71	163.19%	22
Training	Hrs = (-21454) + 1.77 * ESLOC	87.23%	44,520.65	29,827.57	67.00%	20
Air Force	Hrs = 32794 + 0.4897 * ESLOC	48.11%	82,052.78	66,075.85	80.53%	74
Army	Hrs = 1.412e+004 + 0.6038 * ESLOC	41.42%	62,212.15	94,404.19	151.75%	277
Navy	Hrs = (-1.717e+004) + 1.565 * ESLOC	69.79%	106,413.49	176,598.70	165.96%	169
BAE	Hrs = 15537 + 0.5101 * ESLOC	29.93%	33,055.15	30,184.73	91.32%	55
Boeing	Hrs = (-53651) + 1.332 * ESLOC	76.38%	255,857.68	259,834.34	101.55%	37
General Dynamics	Hrs = (-3426) + 1.11 * ESLOC	65.47%	48,343.55	59,073.00	122.19%	105
Lockheed Martin	Hrs = 7264 + 0.5217 * ESLOC	48.23%	55,082.13	62,147.02	112.83%	63
Northup Grumman	Hrs = 3.628e+004 + 0.2429 * ESLOC	21.18%	61,446.55	68,096.08	110.82%	91
Raytheon	Hrs = 2.661e+004 + 0.4418 * ESLOC	32.47%	44,073.50	38,539.34	87.44%	124

TABLE 4: KEY STATISTICS FOR PRODUCTIVITY: PEAK STAFF

Productivity = Peak Staff

Database	Equation	Adjusted R^2	Mean	Standard Error	Coefficient of Variation (Fit Space)	Obs
Overall	Productivity = 3.072 + (-0.01106) * Peakstaff	-0.17%	2.82	23.23	822.44%	475
Incremental	Productivity = 2.362 + (-0.02111) * Peakstaff	1.80%	1.98	2.05	103.92%	90
Sprial	Productivity = 6.288 + (-0.07778) * Peakstaff	-0.56%	4.58	42.49	928.21%	140
Waterfall	Productivity = 3.771 + (-0.03461) * Peakstaff	0.67%	3.19	5.87	184.30%	95
Ada	Productivity = 3.453 + (-0.03974) * Peakstaff	3.30%	2.66	4.11	154.86%	77
C	Productivity = 3.869 + (-0.01069) * Peakstaff	-0.35%	3.62	31.08	858.25%	264
Java	Productivity = 1.168 + (-0.002139) * Peakstaff	-1.08%	1.11	1.49	134.62%	77
Avionics	Productivity = 3.124 + (-0.02194) * Peakstaff	-2.23%	2.70	4.21	155.92%	32
C4I	Productivity = 1.631 + (-0.04009) * Peakstaff	9.68%	1.11	1.27	114.49%	42
Planning	Productivity = 5.9 + (-0.05403) * Peakstaff	15.45%	3.20	4.00	124.92%	7
Simulation	Productivity = 1.644 + (-0.006277) * Peakstaff	-3.24%	1.52	1.31	86.01%	21
Training	Productivity = 0.6042 + 0.03435 * Peakstaff	11.01%	0.94	0.56	59.60%	20
Air Force	Productivity = 1.431 + (-0.005509) * Peakstaff	-1.10%	1.31	1.73	131.47%	73
Army	Productivity = 3.633 + (-0.01036) * Peakstaff	-0.41%	3.36	33.21	988.05%	229
Navy	Productivity = 3.281 + (-0.02415) * Peakstaff	0.66%	2.88	5.21	180.99%	164
BAE	Productivity = 0.9273 + (-0.002348) * Peakstaff	-4.04%	0.90	0.73	81.21%	26
Boeing	Productivity = 1.028 + 0.0009291 * Peakstaff	-1.55%	1.11	1.00	90.14%	35
General Dynamics	Productivity = 1.446 + (-0.005) * Peakstaff	-1.04%	1.37	1.83	133.43%	77
Lockheed Martin	Productivity = 1.169 + (-0.01115) * Peakstaff	1.95%	0.96	1.20	125.17%	60
Northup Grumman	Productivity = 0.9979 + 8.072e-005 * Peakstaff	-1.51%	1.00	0.76	76.26%	68
Raytheon	Productivity = 3.843 + (-0.04592) * Peakstaff	0.11%	3.28	5.59	170.30%	108

So, even though both of these relationships do express the predicted relationships, in most cases the goodness of fit and standard deviation are very poor; indicating that each simple equation by itself would be a poor method to predicting cost estimates for a software development program.

V. Equation Details

Looking at the graph of statistical significance, it can be observed that several of the multivariable equations are statistically significant for the majority of the datasets. This indicates these equations could be used to generate a rough order of magnitude (ROM) estimate for those instances where the results are statistically significant. The following shows the detailed results from the Java dataset for three equations used to find the total duration (in hours) of a project[3]:

$$\text{Duration} = -56355 - 803.1 * \text{Peak staff} + 1.738 * \text{ESLOC} \qquad (3)$$

R^2 = 94.18%   Standard Error = 89,099.42 Observations = 77

$$\text{Duration} = -1.073e{+}004 + 1.633 * \text{ESLOC} - 5.531e{+}004 * \text{CMMI} \qquad (4)$$

R^2 = 94.14%   Standard Error = 79,700.04 Observations = 99

$$\text{Duration} = -47943 - 650.6 * \text{Peak staff} + 1.551 * \text{ESLOC} + 4.917 * \text{Software Req} \qquad (5)$$

R^2 = 94.92%   Standard Error = 83,244.62   Observations = 77

---

[3] See the appendix for a summary of the key statistics of each equation for all datasets.

One potential problem to consider is that the R^2 may be deceptively high for each equation due to the correlation between the independent variables; especially between ESLOC and peak staff[4]. These two variables do a much better job at generating a good R^2, however, their high level of correlation (approximately 0.83 for the Java dataset) indicates a strong level of multicollinearity. The following table shows the correlation matrix for the Java software language dataset:

TABLE 5: CORRELATION MATRIX

	ESLOC	Productivity	Software Requirements	Peak Staff	CMMI	Duration
ESLOC	1.0000	-0.0101	0.7433	0.8322	0.2299	0.9684
Productivity	-0.0101	1.0000	0.0491	-0.0496	-0.0103	0.0609
Software Requirements	0.7433	0.0491	1.0000	0.5858	-0.5449	0.7830
Peak Staff	0.8322	-0.0496	0.5858	1.0000	0.0425	0.7844
CMMI	0.2299	-0.0103	-0.5449	0.0425	1.0000	0.1537
Duration	0.9684	0.0609	0.7830	0.7844	0.1537	1.0000

The matrix shows there is a high level of correlation between both ESLOC/Peak Staff, as well as ESLOC/Software Requirements. It also shows Duration has a strong relationship with ESLOC, Software Requirements, and Peak Staff, demonstrating that these three variables would be good predictors for Duration.

The presence of multicollinearity was not accounted for in any of the analysis. However, the presence of multicollinearity could be cause to use with the simple model (Duration = ESLOC) as opposed to using a more complicated multivariable model.

VI. Software Estimating: Problems

Perhaps the central problem facing software cost estimating is the difficulty to accurately estimate a software development program's size; especially early in the development process (National Aeronautics and Space Administration, 2002). This is caused by several reasons, primarily the creep in requirements, which causes initial size estimates to be ineffective (National Aeronautics and Space Administration, 2002). Generally, the level of uncertainty concerning the program's requirements is a typical estimating problem (Connolly & Dean, 1997). It is imperative that there is a common understanding between not only management and employees, but the customers as well (in the case of defense software applications, the DOD and the contractor).

Since the data shows ESLOC is the most consistent measure of duration (i.e. it is consistently statistically significant), one possible solution is to apply a growth factor to the initial ESLOC estimate to account for the uncertainty problem (Dewberry, 2009). The following growth factors were estimated using data from a large sample and filtered based on the organization's CMMI ratings and software application type (Dewberry, 2009). The following table provides a summary of these growth factors:

---

[4] The number of observations fluctuates due to the missing data points in the dataset.

TABLE 6: ESLOC GROWTH FACTORS

		Segment		
		Air	Ship	Shore
CMMI	3	1.70	1.60	1.50
	4	1.40	1.30	1.20
	5	1.20	1.18	1.10

The following equation was developed (with the following key statistics) for incrementally developed programs written in C++ (Dewberry, 2009):

Duration (Hours) = 41,589 + 0.5817* ESLOC

R^2 = 63.89% Standard Deviation = 69994.54 Observations = 30

The R^2 and other key statistical measurements are much better than the previous equation results for the C dataset; indicating these were better results than in the datasets used for the initial analysis. This could be due to dividing the data into smaller datasets; i.e. the Dewberry model filters the data twice instead of once. While this method does lower the number of data points available, the statistics are much more compelling, indicating a better ensuing cost estimate.

VII. Conclusion

In order to obtain accurate cost estimates for software development programs, it is best if an organization collects data regarding variables that are most effective at estimating the cost (specifically, program size, peak staff, and software requirements; which seem to be the most effective variables to determine a CER for the program). By collecting data and maintaining a company/program specific database, a simple and effective relationship can be used to generate a ROM cost estimate for software development costs.

However, due to changing user needs, it might be impossible to completely eliminate software cost overruns. Until a complete understanding of what the nebulous software development project is at the beginning of the process, there is always a high probability costs will overrun the original estimates. In order to reduce this problem, an estimate can include complexity factors either on the ESLOC (as in the Dewberry model) or via Monte Carlo simulations and S-curves.

VIII. Appendix

Table 7: Equation/Dataset Matrix, Linear Regressions

	Duration = peak staff + ESLOC + productivity	Duration = ESLOC	Duration = Productivity	Productivity = peak staff	Productivity = ESLOC	Productivity = Peak staff + ESLOC	Productivity = Peak staff + SW Requirements	Productivity = ESLOC + CMMI	Productivity = SW Requirements + CMMI	Productivity = Peak Staff + CMMI + ESLOC	Productivity = Peak Staff + CMMI + SW Requirements	Duration = Productivity + SW Requirements	Duration = ESLOC + CMMI	Duration = Productivity + CMMI	Duration = ESLOC + peak staff	Duration =CMMI + ESLOC + Peak Staff	Duration = ESLOC + SW Req + Peak staff	Duration = SW req + CMMI + ESLOC + Peak Staff
Overall	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Incremental	90	90	X	80	90	X	80	X	X	X	X	X	X	X	90	X	X	X
Sprial	X	90	X	X	X	X	X	X	X	X	X	X	X	X	80	X	80	X
Waterfall	X	90	X	X	90	X	X	X	X	X	X	X	X	X	90	X	90	X
Ada	X	90	X	90	80	X	X	X	90	X	X	X	90	X	X	X	X	X
C	X	90	X	X	X	X	X	X	X	X	X	X	80	X	90	X	90	X
Java	80	90	X	X	X	X	X	X	X	X	X	X	90	X	80	X	80	X
Avionics	X	90	80	X	80	X	X	80	90	X	X	90	90	X	X	X	X	X
C4I	90	90	90	90	90	X	80	X	X	X	X	90	90	X	90	80	90	X
Planning	X	90	X	X	X	X	X	90	NA	NA	NA	X	X	90	90	NA	X	NA
Simulation	X	80	NA	X	90	X	X	X	X	X	X	X	X	X	X	X	X	X
Training	90	90	90	90	90	X	X	X	X	80	80	90	90	90	80	X	90	X
Air Force	X	90	X	X	90	X	X	X	X	X	X	80	X	X	90	X	90	X
Army	X	90	X	X	X	X	X	X	X	X	X	X	90	X	90	X	90	X
Navy	90	90	X	80	90	X	X	80	X	X	X	X	X	X	90	90	90	X
BAE	X	90	80	X	90	90	X	80	NA	NA	NA	80	X	80	X	NA	90	NA
Boeing	90	90	X	X	X	X	X	X	X	X	X	X	80	X	90	80	90	80
General Dynamics	X	90	X	X	80	X	X	90	NA	NA	NA	X	X	X	90	NA	80	NA
Lockheed Martin	X	90	X	80	90	X	X	90	X	X	X	X	80	X	X	X	X	X
Northup Grumman	90	90	90	X	80	90	X	NA	NA	NA	NA	90	NA	NA	X	NA	X	NA
Raytheon	X	90	X	X	90	X	X	X	X	X	X	X	X	X	90	X	90	X

X = not statistically significant at the 80% confidence level

NA = Not enough variation in the CMMI dummy variable to run the regression[5]

80 = the F-stat and all t-stats are statistically significant at the 80% confidence level

90 = the F-stat and all t-stats are statistically significant at the 90% confidence Level

Definition of Dependent and Independent Variables

Dependent Variables:

---

[5] As a possible correction for this error and to improve the resulting equations, it is possible that the CMMI dummy variable could be 1 for only those cases where the CMMI level was reported at 4 or 5, and 0 for all other cases.

Duration (Hours): A measure of total development effort; the number of staff-hours that were worked for each observation. Duration is thought of as an interval [T_{start}, T_{finish}]; or [0,t_p] where zero is the start and t_p represents the "p" number of increments necessary for completion (hours are used in our analysis). Total Effort is directly related to program duration, in other words, the sum of all people laboring to complete the task over time "t". The total time to develop SW includes actual development time, understanding, incorporating, changing, and verifying any legacy software.

Productivity (Hours/ESLOC): This is a measure of the rate in which the software can be coded. It shows how many hours it takes to create one ESLOC.

Quantitative Independent Variables:

ESLOC: Source Lines of Code (SLOC) is the delivered size of the product developed, not including any code that was needed to assist development but was not delivered (such as temporary stubs, test scaffoldings, or debug statements). Equivalent Source Lines of Code (ESLOC) is a measure of the program's size, adjusted to account for reuse levels. Mathematically, ESLOC was calculated with the following formula:

ESLOC = New Code + 0.05 Reused Code + 0.50 Modified Code

Size, as measured by ESLOC, is considered to consistently and reasonably represent the work that must be done. Other commonly used size units include function points and algorithms.

Software Requirements: The actual number of software requirements. This does not include count requirements concerning external interfaces not under that project's control.

Peak Staff: This item refers to the actual peak team size, measured in full-time equivalent staff and includes only direct labor.

Productivity (Hours/ESLOC): See above. Definition does not change.

Dummy Independent Variables:

Contractor Maturity Rating: Reports the characterization of the developer's software process maturity using a methodology such as the Software Engineering Institute (SEI) software Capability Maturity Model Integration (CMMI). CMMI is a process improvement model that is based on the principle of achieving continuous improvement through measurement (Defense Acquisition University, 2008). The CMMI provides a framework that is used to measure both the maturity of an organization's software processes as the basis for long-term internal process improvement efforts by the developer and evaluate the developer's software process capability for the purposes of contract award or risk assessment by the acquirer (Defense Acquisition University, 2008).

Table 8: EQUATION 1 SUMMARY

Duration = ESLOC and Peak Staff

Database	Equation	Adjusted R^2	Mean	Standard Error	Coefficient of Variation (Fit Space)	Obs
Overall	Hrs = 69157 + (-64.35) * Peakstaff + 0.08789 * ESLOC	-0.21%	74,901.25	220,379.92	294.23%	436
Incremental	Hrs = 2.057e+004 + 1498 * Peakstaff + 0.5836 * ESLOC	51.34%	99,119.06	67,754.84	68.36%	90
Sprial	Hrs = 5271 + 1887 * Peakstaff + 0.02586 * ESLOC	25.17%	49,695.26	73,846.11	148.60%	140
Waterfall	Hrs = 1.402e+004 + 2373 * Peakstaff + 0.2232 * ESLOC	49.44%	70,231.67	74,708.77	106.37%	95
Ada	Hrs = 61028 + 139.9 * Peakstaff + 0.4022 * ESLOC	8.14%	90,158.18	96,966.83	107.55%	77
C	HRs = 1.577e+004 + 1386 * Peakstaff + 0.2493 * ESLOC	66.66%	69,464.99	67,246.90	96.81%	264
Java	Hrs = (-56355) + (-803.1) * Peakstaff + 1.738 * ESLOC	94.18%	132,566.35	89,099.42	67.21%	77
Avionics	Hrs = 24074 + 3006 * Peakstaff + 0.1128 * ESLOC	32.62%	90,215.18	90,511.42	100.33%	32
C4I	Hrs = 2.337e+004 + (-2880) * Peakstaff + 1.04 * ESLOC	30.29%	43,655.90	50,011.67	114.56%	42
Planning	Hrs = 3181 + (-2727) * Peakstaff + 2.307 * ESLOC	98.45%	279,164.16	32,669.25	11.70%	7
Simulation	Hrs = 220.5 + 4270 * Peakstaff + (-0.01519) * ESLOC	69.69%	81,568.19	77,651.98	95.20%	21
Training	Hrs = (-9258) + (-1833) * Peakstaff + 1.92 * ESLOC	88.29%	44,520.65	28,563.98	64.16%	20
Air Force	Hrs = 9335 + 1695 * Peakstaff + 0.3689 * ESLOC	54.32%	83,176.79	62,076.00	74.63%	73
Army	Hrs = 4302 + 1190 * Peakstaff + 0.3924 * ESLOC	63.30%	70,409.14	80,923.19	114.93%	229
Navy	Hrs = (-2.737e+004) + 2345 * Peakstaff + 1.061 * ESLOC	76.18%	89,808.67	123,622.41	137.65%	164
BAE	Hrs = (-906.2) + 3078 * Peakstaff + (-0.01161) * ESLOC	57.11%	36,950.12	26,702.52	72.27%	26
Boeing	Hrs = (-59373) + (-626.8) * Peakstaff + 1.769 * ESLOC	96.21%	267,342.60	106,570.85	39.86%	35
General Dynamics	Hrs = (-9568) + 826.1 * Peakstaff + 0.9805 * ESLOC	67.01%	53,641.82	60,701.51	113.16%	93
Lockheed Martin	Hrs = 9665 + (-153.9) * Peakstaff + 0.535 * ESLOC	47.42%	56,730.84	63,305.17	111.59%	61
Northup Grumman	Hrs = 1.432e+004 + 2241 * Peakstaff + 0.03511 * ESLOC	45.55%	82,229.94	57,955.37	70.48%	68
Raytheon	Hrs = 2.07e+004 + 1579 * Peakstaff + 0.2504 * ESLOC	43.83%	50,602.91	35,113.84	69.39%	108

Table 9: EQUATION 2 SUMMARY

Duration = ESLOC and CMMI

Database	Equation	Adjusted R^2	Mean	Standard Error	Coefficient of Variation (Fit Space)	Obs
Overall	Hrs = 6.766e+004 + 0.05694 * ESLOC + 1324 * CMMI	-0.19%	72,770.47	199,557.47	274.23%	568
Incremental	Hrs = 6741 + 0.7092 * ESLOC + 3.068e+004 * CMMI	46.23%	99,119.06	71,226.25	71.86%	90
Sprial	Hrs = 48446 + 0.03515 * ESLOC + (-3548) * CMMI	1.38%	49,695.26	84,776.48	170.59%	140
Waterfall	Hrs = 3.02e+004 + 0.3294 * ESLOC + 6898 * CMMI	25.24%	58,977.82	84,241.72	142.84%	118
Ada	Hrs = 65667 + 2.202 * ESLOC + (-126133) * CMMI	58.77%	120,129.25	196,549.73	163.62%	81
C	HRs = 3057 + 0.5167 * ESLOC + 2.302e+004 * CMMI	36.74%	66,193.25	88,986.63	134.43%	291
Java	Hrs = (-1.073e+004) + 1.633 * ESLOC + (-5.531e+004) * CMMI	94.14%	104,981.83	79,700.04	75.92%	99
Avionics	Hrs = 169034 + 0.4547 * ESLOC + (-128218) * CMMI	30.77%	90,215.18	91,749.93	101.70%	32
C4I	Hrs = (-1.175e+004) + 0.6473 * ESLOC + 3.308e+004 * CMMI	26.64%	42,640.65	51,009.74	119.63%	43
Planning	Hrs = (-11692) + 1.702 * ESLOC + (-1.319e+004) * CMMI	92.06%	73,784.18	47,737.99	64.70%	29
Simulation	Hrs = 193701 + (-0.008561) * ESLOC + (-150596) * CMMI	15.91%	82,848.86	126,330.11	152.48%	22
Training	Hrs = 230324 + 0.7532 * ESLOC + (-225129) * CMMI	94.70%	44,520.65	19,207.73	43.14%	20
Air Force	Hrs = 27405 + 0.4895 * ESLOC + 5798 * CMMI	47.41%	82,052.78	66,522.96	81.07%	74
Army	Hrs = (-9346) + 0.6066 * ESLOC + 2.811e+004 * CMMI	41.95%	62,212.15	93,969.92	151.05%	277
Navy	Hrs = 3311 + 1.566 * ESLOC + (-2.458e+004) * CMMI	69.69%	106,413.49	176,889.61	166.23%	169
BAE	Hrs = 8668 + 0.5041 * ESLOC + 7485 * CMMI	28.81%	33,055.15	30,424.38	92.04%	55
Boeing	Hrs = (-120319) + 1.299 * ESLOC + 137673 * CMMI	77.42%	255,857.68	254,013.85	99.29%	37
General Dynamics	Hrs = (-1.118e+004) + 1.105 * ESLOC + 8820 * CMMI	65.20%	48,343.55	59,604.80	122.67%	105
Lockheed Martin	Hrs = (-3.109e+004) + 0.5424 * ESLOC + 4.101e+004 * CMMI	49.63%	55,082.13	61,303.56	111.29%	63
Northup Grumman	NA	NA	NA	NA	NA	NA
Raytheon	Hrs = 4.257e+004 + 0.442 * ESLOC + (-1.609e+004) * CMMI	32.01%	44,073.50	38,670.83	87.74%	124

Table 10: EQUATION 3 SUMMARY

Duration = ESLOC, Software Requirements, and Peak Staff

Database	Equation	Adjusted R^2	Mean	Standard Error	Coefficient of Variation (Fit Space)
Overall	Hrs = 70016 + (-65.23) * Peakstaff + 0.11 * ESLOC + (-1.599) * SWReq	-0.46%	76,128.36	226,814.12	297.94%
Incremental	Hrs = 1.668e+004 + 575.5 * Peakstaff + 0.5557 * ESLOC + 33.4 * SWReq	62.98%	101,964.07	59,797.26	58.65%
Sprial	Hrs = 5782 + 1602 * Peakstaff + 0.02444 * ESLOC + 4.837 * SWReq	27.55%	49,598.88	73,233.18	147.65%
Waterfall	Hrs = 1.824e+004 + 2376 * Peakstaff + 0.3835 * ESLOC + (-35.62) * SWReq	51.63%	73,162.41	74,011.47	101.16%
Ada	Hrs = 6.671e+004 + (-257.6) * Peakstaff + 0.3137 * ESLOC + 5.733 * SWReq	10.18%	92,465.67	96,203.91	104.04%
C	HRs = 1.375e+004 + 1375 * Peakstaff + 0.2225 * ESLOC + 7.005 * SWReq	66.91%	70,259.89	67,686.15	96.34%
Java	Hrs = (-47943) + (-650.6) * Peakstaff + 1.551 * ESLOC + 4.917 * SWReq	94.92%	132,566.35	83,244.62	62.79%
Avionics	Hrs = 118684 + 2433 * Peakstaff + 0.1658 * ESLOC + (-99974) * CMMI	39.78%	90,215.18	85,565.46	94.85%
C4I	Hrs = 8798 + (-1986) * Peakstaff + 0.6932 * ESLOC + 52.33 * SWReq	44.92%	44,553.49	44,793.71	100.54%
Planning	Hrs = (-4875) + (-2694) * Peakstaff + 2.207 * ESLOC + 41.39 * SWReq	98.77%	279,164.16	29,070.71	10.41%
Simulation	Hrs = 1.157e+004 + 4579 * Peakstaff + 0.01461 * ESLOC + (-37.96) * SWReq	70.94%	81,568.19	76,031.32	93.21%
Training	Hrs = (-1.45e+004) + (-2084) * Peakstaff + 2.673 * ESLOC + (-83.14) * SWReq	94.33%	44,520.65	19,879.55	44.65%
Air Force	Hrs = 1.608e+004 + 1893 * Peakstaff + 0.4641 * ESLOC + (-44.6) * SWReq	60.21%	87,054.41	59,105.91	67.90%
Army	Hrs = (-3674) + 1092 * Peakstaff + 0.3353 * ESLOC + 28.14 * SWReq	66.31%	70,453.89	78,129.77	110.89%
Navy	Hrs = (-1.657e+004) + 1576 * Peakstaff + 0.8423 * ESLOC + 9.615 * SWReq	79.75%	92,315.62	115,566.50	125.19%
BAE	Hrs = (-1539) + 3093 * Peakstaff + 0.3319 * ESLOC + (-46.4) * SWReq	68.83%	36,950.12	22,761.87	61.60%
Boeing	Hrs = (-5.912e+004) + (-381.9) * Peakstaff + 1.577 * ESLOC + 4.301 * SWReq	96.64%	267,342.60	100,359.41	37.54%
General Dynamics	Hrs = (-1.382e+004) + 1226 * Peakstaff + 0.9801 * ESLOC + (-9.058) * SWReq	70.61%	57,464.18	60,841.91	105.88%
Lockheed Martin	Hrs = 10196 + 9.751 * Peakstaff + 0.6233 * ESLOC + (-24.58) * SWReq	51.17%	57,669.68	61,303.30	106.30%
Northup Grumman	Hrs = 9787 + 2193 * Peakstaff + 0.01629 * ESLOC + 7.403 * SWReq	46.54%	82,229.94	57,423.39	69.83%
Raytheon	Hrs = 2.385e+004 + 1952 * Peakstaff + 0.2426 * ESLOC + (-16.8) * SWReq	47.71%	52,418.34	34,294.10	65.42%

IX. Bibliography

Connolly, T., & Dean, D. (1997). Decomposed Versus Holistic Estimates of Effort Required for Software Writing Tasks. *Management Science* , 1029-1045.

Defense Acquisition University. (2008, October 7). *Basic Systems Acquisition Management* . 2008, Virginia, USA: Defense Acquisition Workforce Improvement Act.

Defense Cost Resource Center. (2008, October). Software Resource Data Report Database. Department of Defense.

Dewberry, C. (2009, August 24). Software CER Documentation. (J. Leotta, Interviewer)

Hoegl, M., & Georg Gemuenden, H. (2001). Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence. *Organization Science* , 435-449.

Jensen, R. W., Putnam, L. H., & Roetzheim, W. (2006, February). *Crosstalk, the Journal of Defense Software Engineering*. Retrieved August 25, 2009, from Software Estimating Models: Three Viewpoints:

http://www.stsc.hill.af.mil/crosstalk/2006/02/0602JensenPutnamRoetzheim.html

Maxwell, K., & Van Wassenhove, L. (1999). Performance Evaluation of General and Company Specific Models in Software Development Effort Estimating. *Management Science* , 787-803.

Minkiewicz, A. (2008). *Computer Aid Inc*. Retrieved August 25, 2009, from The Evolution of Software Estimating: www.compaid.com/caiinternet/ezine/Estimating-AM.pdf

National Aeronautics and Space Administration. (2002, May 17). *Parametric Cost Estimating Handbook*. Retrieved August 23, 2009, from Chapter 5: Software Parametric Cost Estimating: http://cost.jsc.nasa.gov/PCEHHTML/pceh.htm

Ross, M. A. (2008). Next Generation Software Estimating Framework: 25 Years Later and Thousands of Projects Later. *Society of Cost Estimating and Analysis- International Society of Parametric Analysis* , 7-15.

Wu, L. (1997, May 4). *University of Calgary*. Retrieved August 25, 2009, from Comparison of the Software Cost Estimating Models: http://www.compapp.dcu.ie/~renaat/ca421/LWu1.html