NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

SYSTEMS ENGINEERING CAPSTONE REPORT

RECOMMENDATIONS FOR IMPROVING SOFTWARE COST ESTIMATION IN DOD ACQUISITION

by

Steven F. Daley, Martin L. Hogan, Joshua D. Stokes,
and Suzanne M. Vermeulen

September 2022

Advisor:	Raymond J. Madachy
Co-Advisor:	John M. Green

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2022	**3. REPORT TYPE AND DATES COVERED** Systems Engineering Capstone Report	
4. TITLE AND SUBTITLE RECOMMENDATIONS FOR IMPROVING SOFTWARE COST ESTIMATION IN DOD ACQUISITION			**5. FUNDING NUMBERS**
6. AUTHOR(S) Steven F. Daley, Martin L. Hogan, Joshua D. Stokes, and Suzanne M. Vermeulen			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Acquisition initiatives within the Department of Defense (DOD) are becoming increasingly reliant on software. While the DOD has ample experience in estimating costs of hardware acquisition, expertise in estimating software acquisition costs is lacking. The objective of this capstone project is to summarize the current software cost estimating methods, analyze existing software cost estimating models, and suggest areas and methods for improvement. To accomplish this, surveys were conducted to gather program cost data, which was run through existing cost estimating models. From here, the outputs were compared to actual program costs. This established a baseline for the effectiveness of existing methods and guided suggestions for areas of improvement. The Software Resource Data Reports (SRDR) data used seemed to have spurious data reporting from at least one source, and the base cost estimation models were not found to be sufficiently accurate in our study. The capstone finds that calibrating the cost models to the data available improved those models dramatically. In all, the capstone recommends performing data realism checks upon SRDR submissions to ensure data accuracy and calibrating cost models for each contractor with the available data before using them to estimate DOD Acquisition costs.

14. SUBJECT TERMS			**15. NUMBER OF PAGES**
software, cost estimation modeling			91
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

RECOMMENDATIONS FOR IMPROVING SOFTWARE COST ESTIMATION IN DOD ACQUISITION

Steven F. Daley, Martin L. Hogan, Joshua D. Stokes, and Suzanne M. Vermeulen

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2022

Lead Editor: Suzanne M. Vermeulen

Reviewed by:	
Raymond J. Madachy	John M. Green
Advisor	Co-Advisor

Accepted by:
Oleg A. Yakimenko
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Acquisition initiatives within the Department of Defense (DOD) are becoming increasingly reliant on software. While the DOD has ample experience in estimating costs of hardware acquisition, expertise in estimating software acquisition costs is lacking. The objective of this capstone project is to summarize the current software cost estimating methods, analyze existing software cost estimating models, and suggest areas and methods for improvement. To accomplish this, surveys were conducted to gather program cost data, which was run through existing cost estimating models. From here, the outputs were compared to actual program costs. This established a baseline for the effectiveness of existing methods and guided suggestions for areas of improvement. The Software Resource Data Reports (SRDR) data used seemed to have spurious data reporting from at least one source, and the base cost estimation models were not found to be sufficiently accurate in our study. The capstone finds that calibrating the cost models to the data available improved those models dramatically. In all, the capstone recommends performing data realism checks upon SRDR submissions to ensure data accuracy and calibrating cost models for each contractor with the available data before using them to estimate DOD Acquisition costs.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

AAF Adaptation Adjustment Factor

ACAT Acquisition Category

AppType Application Type

C&C Command and Control

CER Cost Estimating Relationship

COCOMO Constructive Cost Model

COSYSMO Constructive Systems Engineering Cost Model

CSDR Cost and Software Data Report

CS&CSS Combat Support & Combat Service Support

CUI Controlled Unclassified Information

DFARS Defense Federal Acquisition Regulation Supplement

DoD Department of Defense

EM Effort Multiplier

ESLOC Equivalent Software Lines of Code

FAR Federal Acquisition Regulation

FP Force Projection

FTE Full-Time Equivalent

GCS Ground Combat Systems

GV Ground Vehicle

IGCE	Independent Government Cost Estimate
KESLOC	Kilo Equivalent Software Lines of Code
KSLOC	Kilo Software Lines of Code
KTR	Contractor
MMRE	Mean Magnitude Relative Error
MRE	Magnitude Relative Error
NASA	National Aeronautics and Space Administration
OpEnv	Operating Environment
PEO	Program Executive Office
PM	Program Management
P-M	Person-Months
PMO	Program Management Office
PRED(X)	Prediction Accuracy
RE	Relative Error
SEER-SEM	Software Evaluation and Estimation of Resources - Software Estimating Model
SF	Scale Factor
SLOC	Software Lines of Code
SME	Subject Matter Expert
SRDR	Software Resources Development Report
SSTS	Sustainment System Technical Service
STS	System Technical Service
WBS	Work Breakdown Structure

Executive Summary

The Department of Defense (DoD) of today is employing and relying upon software products to achieve their missions at an ever-increasing rate. As systems become more digitized and network-centric, the need for software development, integration, and maintenance have become commonplace throughout the organization. Estimating the cost and effort required for these projects in a standardized way has been challenging for Program Management Offices (PMOs). Cost estimation of a software project is necessary for schedule planning, budgeting, staffing, analysis of alternative proposals, and untold other systems engineering and program management functions. It has been a focus of study for decades, and multiple cost models have been developed and refined out of that study, which aid in accurate and timely estimates being produced and validated.

Software efforts have become a major cost driver for many Army programs in recent years, and that trend is expected to continue. This capstone project focuses on software cost estimating for efforts within the United States Army Program Executive Office (PEO) Ground Combat Systems (GCS) PMOs. Software cost estimates provided by contractors were analyzed using selected established estimating methods and compared to the actual cost data of the completed projects, to determine which method is most useful.

A background research and literature review was conducted by the team to establish a baseline for the current subject matter expertise within this research area. This provided a backdrop for the project and informed the team on possible avenues for study. It ultimately led the team to choosing the methods evaluated in this capstone project.

Software cost and effort data were provided for study from various programs within PEO GCS and PEO Combat Support & Combat Service Support (CS&CSS). The DoD collects software data from contractors on Software Resources Development Reports (SRDRs), which are standard forms that require the same data from each vendor. Those were provided by the PMOs and that data was scrubbed for completeness and usability to determine prime candidates for study. After that scrub, the finalists were compared to actual project data to determine which would provide for the most complete and in-depth analysis. This resulted in a total of 19 unique software efforts to serve as data points for analysis, from

two separate contractors. This data was run through three software cost estimating models, the Constructive Cost Model (COCOMO) II (Boehm 2009), the Command and Control (C&C) Cost Estimating Relationship (CER) (Clark 2015), and the Ground Vehicle (GV) CER (Clark 2015) to produce estimates from the data. Those estimates were then compared to the actual cost data for those efforts provided by the PMOs to determine the veracity and usefulness of each method.

The analysis was performed with each method uncalibrated and then calibrated for each individual contractor. The results were such that none of the methods were very accurate without calibration, making the need for tailored calibration obvious. After each model was calibrated for the contractor and effort being examined, the results were much closer to the actuals being reported by the programs. The calibrated COCOMO II model proved to be the best fit model overall, with MMRE values of 76.2% (combined data set), 93.3% (Contractor 1), and 66.7% (Contractor 2). The calibrated COCOMO II model proved to be the best fit again with the prediction accuracies, with PRED(20) = 15.8%, PRED(30) = 21.1%, and PRED(50) = 52.6% for the combined data set. For Contractor 1, the calibrated COCOMO II model and the calibrated CER had identical prediction accuracies of PRED(20) = 33.3%, PRED(30) = 44.4%, and PRED(50) = 66.7%. For Contractor 2, there was again a tie between the calibrated COCOMO II model and the calibrated CER, with PRED(20) = 30.0%, PRED(30) = 30.0%, and PRED(50) = 60.0%. If the COCOMO scale and cost factors were adjusted away from nominal, there would likely be more of a prediction accuracy difference noted between calibrated COCOMO II and the calibrated CER. During the analysis, data anomalies were observed and their possible provenance was explored; exploring these anomalies is worth noting because outliers can be expected in real-world use.

The team arrived at a number of conclusions after this complete analysis. As noted throughout this capstone paper, these models are sensitive to the type of software effort being examined, the system the software is being integrated upon, and the project environment, and thus all require tailoring and calibration. Therefore, the results indicate that calibration may be more important than the choice of estimation model. It is also worth mentioning that the process of performing this analysis provided a number of additional insights that resulted in recommendations for PMOs moving forward. These recommendations include the use of standardized checklists to focus the process, calibrating the models as appropriate for each effort and vendor, building internal expertise in scale and cost driver factors inherent in each method, improving and standardizing data collected on a SRDR, refining requirements, adjusting regulatory requirements, and establishing internal centers of excellence.

References

B. W. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. J. Reifer, and B. Steece, *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ: Prentice Hall Press, 2009.

B. Clark and R. Madachy, *Software Cost Estimation Metrics Manual for Defense Systems*. Haymarket, VA: Software Metrics Inc., 2015.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

We would like to thank our advisors and professors, John Green and Raymond Madachy, for their guidance and support over the capstone coursework and toward the success of our project. We would also like to thank all the other professors at the Naval Postgraduate School, who took the time to teach us and provide the background knowledge and expertise needed for this capstone. We would also like to thank our family, friends, loved ones, and fellow NPS students who provided us with much needed support throughout this master's program. Lastly, we would like to thank our respective Program Executive Offices (PEOs) for providing source data to make this capstone possible.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction, Background, Models Examined, and Literature Review

1.1 Introduction

Software efforts have become a major cost driver for Army programs, particularly in Program Executive Office (PEO) Ground Combat Systems (GCS) programs. As systems become more digitized and network-centric, the requirements for software development, integration, and maintenance have become commonplace throughout the organization. Cost estimating these efforts has been challenging for the program offices, as a standardized cost estimation process based upon best business practices and processes has been inconsistent. Cost estimation methods such as the Constructive Cost Model (COCOMO) II and Department of Defense (DoD) domain-specific Cost Estimating Relationships (CERs) have been developed to perform such cost estimates.

A standardized process for the program management organization will result in consistent, accurate, and timely estimates for these software efforts.

A key enabler for software cost estimation is matching the process to the data available to conduct the estimate. Software Resources Development Reports (SRDRs) are a recent requirement added to software contracts to provide some of this data as well as to track software effort progress with sufficient granularity to determine impacts to cost and schedule.

An optimized software cost estimation process for System Technical Service (STS) contracts will enable the Program Management Office (PMO) to perform best value decisions as well as budget management and forecasting. These management offices continually integrate new requirements with new software developments, maintenance of existing code, and integration of software laden subsystems into the vehicle platform. Depending upon the nature of the effort, the cost estimate may require one or all of these aspects. The resulting cost estimation process must account for all these scenarios as well as the varied data available to conduct the cost estimation.

1.2 Background

This section provides background on the basics of software project cost estimating and briefly describes the methods chosen for this research project. This chapter then provides an overview of the literature reviewed on the topic. The literature review is focused on existing software cost estimating methods and models, used in both industry and DoD applications. It includes a discussion on cost estimating theory and best-fit approaches.

Software cost estimation aims to predict the effort required to develop a software system, software components of a larger hardware system, and maintenance endeavors to update, improve, and secure software systems. The effort predictions generally consist of labor hours required, estimated schedule, and overall cost. Many estimation methods, tools, and models have been proposed over the years, each with a goal of creating the most accurate prediction possible, both for the vendor and the customer. In the DoD and for Army ground combat systems, software cost estimating efforts are generally focused on evaluating vendor proposals to determine which offeror can satisfy requirements at the best value.

In their wide-reaching 2007 paper, "A Systematic Review of Software Development Cost Estimation Studies," Magne Jørgensen and Martin Shepperd [1] explore the breadth and depth of previous scholarly work on the topic of software cost estimation and offer their expert opinions on the value each contributes. As the review identifies 304 software cost estimation papers in 76 journals, it is an excellent primer and introduction to the topic. The authors categorized the papers according to topic of research, how the estimates were arrived at, methods of research, the contexts in which the studies were conducted, and the size and type of data set. They created a library of the cost estimation papers to aid in the identification of relevant estimation research results. The review results combined with background knowledge and subject matter expertise on a particular program or industry provide support for recommendations for future software cost estimation research. These recommendations include increasing the range of topics and industries of the search for relevant studies, conducting a thorough manual search for applicable papers within a narrowly specified set of journals (to ensure completeness), conducting additional studies on estimation methods common to the software industry, and educating researchers and practitioners on how the data set properties can impact the results when evaluating these methods. All of these recommendations aided in guiding this research project and developing this capstone paper.

1.3 Models Examined

As a result of the team's background research and literature review, three CER models were chosen to evaluate software projects within the Army's ground combat system portfolio. A CER is an equation used to estimate cost using a known relationship with one or more independent variables. These include the COCOMO II model, the Command and Control (C&C) CER, and the Ground Vehicle (GV) CER as they were considered applicable to the domain and environment of the work reflected in the SRDRs.

1.3.1 COCOMO II

COCOMO II is a procedural software cost estimation model developed by Barry W. Boehm and colleagues demonstrated in the book "Software Cost Estimation with COCOMO II" [2]. The model uses software scale and cost drivers to refine and describe environmental factors that affect team performance, schedule, and project cost. The "parameters of the model are derived from fitting a regression formula using data from historical projects" [2]. There is also an online tool with an intuitive user interface that simplifies the process for the user (http://softwarecost.org/tools/COCOMO/) [3].

1.3.2 C&C and GV Cost Estimating Relationships

The CERs contained in the "Software Cost Estimation Metrics Manual for Defense Systems" by Bradford Clark and Raymond Madachy [4] use software size as a predictor and effort as the response. They take into account the type of application and the operating environments to develop the CERs and benchmarks. The application type groups are based on similar functional characteristics, while the operating environment groups distinguish and describe the environments in which the applications operate.

1.4 Literature Review

The team conducted an extensive literature search and review effort, mainly utilizing the resources at the Dudley Knox Library, to build a foundation of knowledge from which to launch this capstone research project. The goal of the literature review was to review the previous efforts in the arena of software cost estimating and to provide a backdrop for critical thinking and analysis on the topic. A secondary goal was to guide the team to the narrower focus of software cost estimating in Army ground system program offices. Those sources are summarized below.

The “Software Cost Estimation Metrics Manual for Defense Systems” is a manual edited by Bradford Clark and Raymond Madachy [4], that attempts to improve the software development cost metrics in the DoD. The manual covers the metrics for reporting and analyzing DoD cost data, and works to standardize them for CER creation. It also explains how to use SRDR data that is available to the DoD. We used this data to help with the data analysis from the SRDR data we collected.

“Software Development Effort Estimation: Formal Models or Expert Judgment?” is a paper written by Magne Jørgensen, Barry Boehm, and Stan Rifkin [5]. In this paper, two experts in the field of software development, Magne Jørgensen and Barry Boehm, take opposing views of the best method for software project cost estimation in a debate-style format; formal models vs expert judgment. The aim of this argument is to illustrate in which scenarios, both project and environment based, each method is best suited. In illustrating these points, the paper goes on to dispel three misconceptions. The first is that models are objective. The paper makes the point that models are only as good as their inputs, which are subject to human interpretation and bias. The second misconception is that judgment-based estimation is a highly subjective, nebulous process that cannot be improved. Barry Boehm’s “Wideband Delphi” process and the guidelines suggested in “Practical Guidelines for Expert-Judgment-Based Software Effort Estimation" [6] provide suggestions for ways to improve. The third misconception is that the more advanced an estimation model is, the more accurate it will be. The paper argues that uncomplicated and straightforward models typically perform on par with more advanced models and that more advanced and complicated models are vulnerable to overfitting to unrepresentative or exaggerated data sets.

The “Army Cost Analysis Manual” [7] is written to provide procedures for conducting Army cost analysis. It contains the methods to develop, analyze, and document cost estimates for Army. This manual was used to find current cost estimating techniques used in the Army for comparison during our study.

“Estimating With Use Case Points” by Mike Cohn of Mountain Goat Software [8], is an article about using Use Case Points to develop an estimate. The article assigns weights to the number and complexity of the use cases that were developed for the system, and uses those in a calculation along with data on the number of actors, requirements, and environments to come up with a cost estimate.

“A Study of Software Development Cost Estimation Techniques and Models” by Junaid Rashid, Muhammad Wasif Nisar, Toqeer Mahmood, Amjad Rehman, and Syed Yasser Arafat [9] is a comprehensive look of over 20 software cost estimating methods, to include COCOMO, COCOMO II, Constructive Systems Engineering Cost Model (COSYSMO), and the DoD specific Work Breakdown Structure (WBS) methods. The paper is easily digestible by an inexperienced user, using accessible language and examples that focus on the pros and cons of each. It makes the very astute observation that no one method is a perfect fit and different situations and environments require different methods and most often a combination of methods to create a best fit. The authors also observe that a method or tool should be calibrated for each scenario; a tool developed in one environment will need to be calibrated when used in another.

In his article “What We Do and Don’t Know about Software Development Effort Estimation” [10], Magne Jørgensen makes a case for expert opinion continuing to be the most accurate software project cost estimating method. He highlights that estimation accuracy has not significantly improved since the 1980s. He joins many others in emphasizing that there is no one best model or method for cost estimation, and that some are better suited to certain situations than others. While maintaining that expert opinion is still the most accurate method, he does point out that certain tools and strategies can be used in that effort to increase accuracy and consistency, like using checklists to ensure often over-looked processes are included, tailoring methods to a company or certain types of projects, using historical data as bases, avoiding exposure to “leading” language, using group-based estimation structures, and avoiding too early of estimates based on incomplete information.

The “DoD Cost Estimating Guide” [11] outlines four cost estimating methods: analogy, build-up, extrapolation of actuals, and parametric. Analogy is estimating an effort based upon the known costs of similar systems or efforts. The build-up method is a more detailed method that sums estimates of lower-level elements of the larger, total effort. Extrapolation of actuals estimates future costs of an already started effort using actual costs to date and projecting them across the remaining effort. Parametric estimating involves relating costs to programmatic characteristics through an algebraic equation. The most common method for software cost estimation for ground vehicle systems has historically been the analogy method. Figure 1.1 summarizes the advantages and disadvantages of these methods.

Estimating Method	Advantages	Disadvantages
Analogy	• Applicable before detailed program requirements are known • Can be developed quickly • Completed analogous program inherently includes risk and uncertainty • Based on objective historical data that can be readily communicated and understood	• Relies on a single data source • May require adjustments for risks/opportunities and uncertainties not present in the current program • Technical data required for scaling may be elusive and/or difficult to defend • Subjectivity with technical parameter adjustment factors likely to be introduced • Appropriate analogy may not be available
Build-Up	• Fully documents and addresses exactly what the cost estimate includes • Captures the specific manufacturer's processes and rates • Explicitly reveals the major cost contributors • Provides a basis to check for duplicates and omissions	• May be expensive to implement and time consuming • Less flexible and may not answer many of the what-if questions • New estimates must be built for each alternative • Product specification must be well known and stable • All product and process changes must be reflected in the estimate • Small errors can grow into larger errors through the summation • Elements can easily be omitted or duplicated by accident in large models
Extrapolation of Actuals	• Uses the program actual data to develop the estimate	• Access to sufficient and reliable cost data may be challenging • Changes in accounting, engineering, and manufacturing processes have to be identified and addressed
Parametric	• Versatile and can be derived at any level where the data is available • Supports what-if explorations of design alternatives • Supports cost driver sensitivity analysis • Provides objective measures of statistical significance of each coefficient and of the model as a whole • Provides objective measure of uncertainty (standard error) • Objective measure of the result's probability of exceedance • Derived from objective historical data	• Source data must be consistent, accurate, and properly normalized • Often have to rely on a few data points • Cannot use without fully understanding how it was generated • Must be updated to capture the current cost, technical, and program data • Populating with independent variable values outside the range of the source data leads to increased uncertainty and may produce erroneous results • Complicated relationships may be difficult to defend

Figure 1.1. Basic Estimating Methods. Source: [11]

The introductory article "Actionable Analytics for Software Engineering" [12] emphasizes how intensive research has been conducted and data collected regarding software analytics but the application and use of the results remains lacking. They asked the rhetorical question, "what can we do with all that research?". The authors posit that simple methods can be "as good or even better than more complicated ones as long as they're applicable and work." [12]

In their article "Assuring Software Cost Estimates: Is it an Oxymoron?" [13], Hihn and Tregre recognize the repeated cost growth of software efforts despite many years of cost estimation research and propose applying software assurance methodology to software cost estimates. "In a study of National Aeronautics and Space Administration (NASA) software development projects conducted in the late nineties, the most frequently identified cause (71%) of cost overrun with the largest impact (35% contribution to observed cost growth) was basic failures in planning, estimation control." [2] "One contributor to such inaccuracies is that software engineers and managers continue to perform bottom up estimates with little or no data to support their assumptions and little or no consideration for risk and uncertainty." [5] The authors recommend and further define the following cost estimation best practices: establish a cost estimation infrastructure, perform the estimate, document the estimate, and monitor performance.

In their article "Assuring Software Cost Estimates: Is It an Oxymoron?", Hihn and Tregre [13] suggest applying the software assurance concepts to cost estimating best practices. These include following established guidelines, standards, and processes, allowing tailoring when appropriate, tracing cost estimates to particular development activity, proper training, ensuring proper application of algorithms, conducting peer reviews, and using metrics to expose problematic trends.

The "Software Development Cost Estimating Handbook" by the Software Technology Support Center Cost Analysis Group [14] provides a foundation for performing cost estimates for software development projects. The authors include general process concepts and lay out the major cost drivers for software projects. Of interest is the life-cycle focused activities outlined in the document and a robust description of the variables associated with many of the current software cost estimating models, such as Software Evaluation and Estimation of Resources - Software Estimating Model (SEER-SEM), COCOMO II, and Price-S.

The "Handbook for Software Cost Estimation" by Karen Lum, Michael Bramble, Jairus Hihn, John Hackney, Mori Khorrami, and Erik Monson [15] is a cookbook of sorts for program managers and offices to perform software cost estimates for the various efforts that they may endeavor during their program execution. It details recommended step-by-step processes to perform cost estimates for a variety of situations such as full development, and re-use and modification of code type projects. Cost estimation methods include analogy, subject matter expert opinion, model-based, and "rules-of-thumb." Figure 1.2 describes the general cost estimation steps that are included in the handbook. The handbook goes on to describe the steps and efforts taken during each step, which ultimately leads to a cost estimate, review of the estimate, and maintenance of the estimates. They also note that depending upon the estimate's level in the WBS, the methods often change with the level of data available.

Action	Description	Responsibility	Output Summary
Step 1: Gather and Analyze Software Functional & Programmatic Requirements	Analyze and refine software requirements, software architecture, and programmatic constraints.	Software manager, system engineers, and cognizant engineers.	• Identified constraints • Methods used to refine requirements • Resulting requirements • Resulting architecture hierarchy • Refined software architecture • Refined software functional requirements
Step 2: Define the Work Elements and Procurements	Define software work elements and procurements for specific project.	Software manager, system engineers, and cognizant engineers.	• Project-Specific product-based software WBS • Procurements • Risk List
Step 3: Estimate Software Size	Estimate size of software in logical Source Lines of Code (SLOC).	Software manager, cognizant engineers.	• Methods used for size estimation • Lower level and total software size estimates in logical SLOC
Step 4: Estimate Software Effort	Convert software size estimate in SLOC to software development effort. Use software development effort to derive effort for all work elements.	Software manager, cognizant engineers, and software estimators.	• Methods used to estimate effort for all work elements • Lower level and Total Software Development Effort in work-months (WM) • Total Software Effort for all work elements of the project WBS in work-months • Major assumptions used in effort estimates
Step 5: Schedule the effort	Determine length of time needed to complete the software effort. Establish time periods of work elements of the software project WBS and milestones.	Software manager, cognizant engineers, and software estimators.	• Schedule for all work elements of project's software WBS • Milestones and review dates • Revised estimates and assumptions made
Step 6: Calculate the Cost	Estimate the total cost of the software project.	Software manager, cognizant engineers, and software estimators.	• Methods used to estimate the cost • Cost of procurements • Itemization of cost elements in dollars across all work elements • Total cost estimate in dollars
Step 7: Determine the Impact of Risks	Identify software project risks, estimate their impact, and revise estimates.	Software manager, cognizant engineers, and software estimators.	• Detailed Risk List • Methods used in risk estimation • Revised size, effort, and cost estimates
Step 8: Validate and Reconcile the Estimate Via Models and Analogy	Develop alternate effort, schedule, and cost estimates to validate original estimates and to improve accuracy.	Software manager, cognizant engineers, and software estimators.	• Methods used to validate estimates • Validated and revised size, effort, schedule, and cost estimates.
Step 9: Reconcile Estimates, Budget, and Schedule	Review above size, effort, schedule, and cost estimates and compare with project budget and schedule. Resolve inconsistencies.	Software manager, software engineers, software estimators, and sponsors.	• Revised size, effort, schedule, risk and cost estimates • Methods used to revise estimates • Revised functionality • Updated WBS • Revised risk assessment
Step 10: Review and Approve the Estimates	Review and approve software size effort, schedule, and cost estimates.	The above personnel, software engineer with experience on similar project, line and project management.	• Problems found with reconciled estimates • Reviewed, revised, and approved size, effort, schedule, and cost estimates • Work agreement(s), if necessary
Step 11: Track, Report, and Maintain the Estimates	Compare estimates with actual data. Track estimate accuracy. Report and maintain size, effort, schedule, and cost estimates at each major milestone.	Software manager, software engineers and software estimators	• Evaluation of comparisons of actual and estimated data • Updated software size, effort, schedule, risk and cost estimates • Archived software data

Figure 1.2. Software Estimation Steps. Source: [15]

The current standard and most common software cost estimating practice in Army ground system program offices is the Independent Government Cost Estimate (IGCE). The Federal Acquisition Regulation (FAR) [16] and Defense Federal Acquisition Regulation Supplement (DFARS) [17] provide guidance on IGCEs. This process and resulting document are heavily reliant on analogy and expert opinion. While this is arguably a valid method, as evidenced in the reviewed literature, this discovery during the survey process led the team to question whether this should still be the standard. This resulted in the selection of the above mentioned modeling methods and this capstone research project.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Methodology

This chapter describes the approach used to evaluate different cost estimation methods for Army ground combat vehicle software efforts. It details the current methods used in program offices, a description the of the efforts examined, the source data and metrics used for estimating efforts, the software estimation models examined, assumptions made, and the methodology for evaluating the cost estimation models.

Figure 2.1 illustrates the steps the team followed to collect data, apply assumptions, employ and calibrate cost models, and evaluate the prediction accuracy of the cost models. Each process step in the figure aligns to a section of this methodology. The outputs of each step are outlined and illustrated as inputs used by the next step in the process.

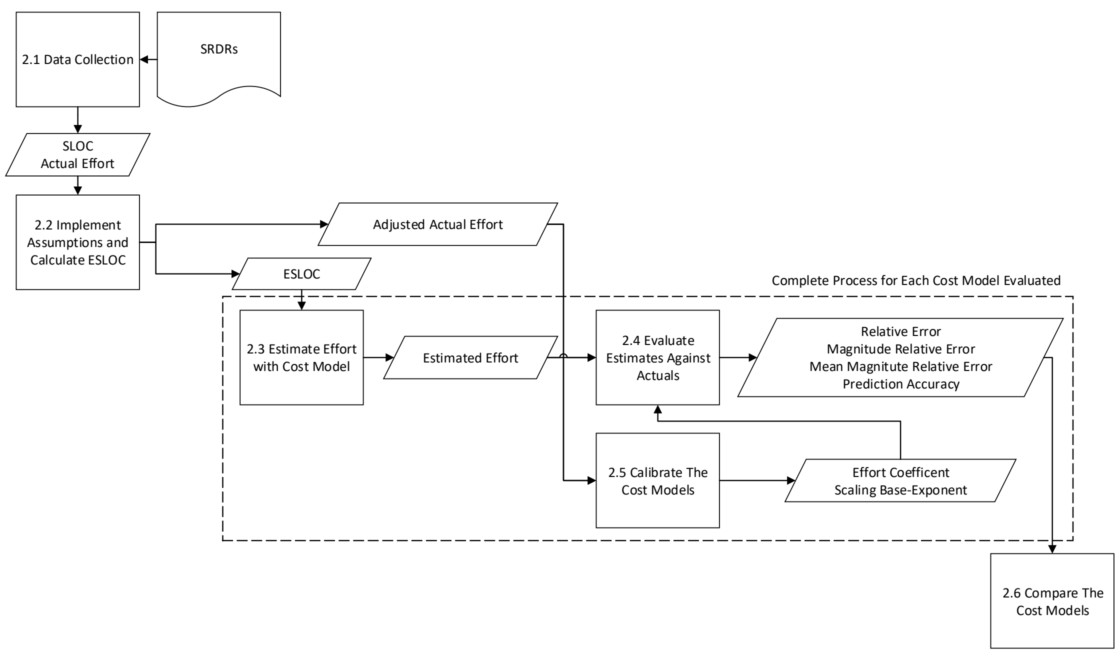

Figure 2.1. Methodology

2.1 Data Collection

The team surveyed Army Program Management and Product Management Offices across PEO GCS and PEO Combat Support & Combat Service Support (CS&CSS) to understand how current software effort estimates are conducted. A large majority of programs use analogous estimation methods; they choose a similar program or effort and apply a complexity coefficient based on Subject Matter Expert (SME) opinions to estimate the costs of the upcoming software project.

In a deep dive into the methods employed by various programs across Program Management (PM) Force Projection (FP), our research revealed initial efforts to break away from the analogy method. One program used a bottoms-up approach combined with engineering experience to estimate hourly rates only. While this method was based in reality, by using existing Full-Time Equivalent (FTE) rates from a similar project, it did not go beyond this point, to examine the estimate costs against Software Lines of Code (SLOC), for example.

Another methodology employed, was the use of TruePlanning software. This is a form of parametric estimating; however, the data involved is contained within a "black box", so a relationship to be drawn between cost and SLOC is also not possible with this method. It is worth noting that much of the research indicates that the DoD has been collecting software related data on programs through their Cost and Software Data Reports (CSDRs) over the years, which would lend to the potential to use analogies and/or CERs in future efforts.

2.1.1 Source Data Metrics for Cost Estimating Model Evaluation

In the spirit of Yang et al's idea that "simple methods can be as good or even better than more complicated ones as long as they're applicable and work" [12] from "Actionable Analytics for Software Engineering", the team utilized SRDRs from programs within PEO GCS. The SRDRs are provided to the Government by contractors as contract deliverables that document a breakdown of software efforts by WBS element and outline the SLOC for the effort, to include a breakdown of new, modified, and reused SLOC, and actual hours spent on each element. This information provides an excellent starting point to conduct rather fast estimates and evaluate the output to actuals for cost estimation methods that employ SLOC as a key input.

2.1.2 Programs That Provided Source Data

As mentioned above, all SRDR information was provided by programs within PEO GCS. This is an important distinction to make because it bounds the data sets by similar application types, i.e. ground combat systems, and limited the number of contractors compared. The idea is that this should help limit any outliers given the similarity of the systems and limited differences in company software development processes and procedures employed to execute the efforts.

As this data is all proprietary data, it was sanitized by removing any identifying information. The contractor names were removed and are now referred to as Contractor 1 and Contractor 2. All programs were renamed by sequential numbers (1,2,3, etc.) and the WBS names were renamed to Projects with sequential letters (A,B,C, etc.).

2.2 Implement Assumptions and Calculate ESLOC

The SRDR data for all the programs reviewed included WBS elements for which there were hours but no SLOC assigned. Given the team was interested in cost evaluation methods for software efforts as a whole, the team chose to include those hours in the evaluation. The hours for these WBS elements were included by adding them to SLOC containing WBS elements of the SRDR using a weighted average. This was done by summing the non-SLOC WBS element hours (labeled support hours), multiplying that total by the percentage of total SLOC a WBS element contributed to the whole effort, and adding the hours to the SLOC containing WBS within that program (shown in Equation 2.1.)

$$Adjusted\ Hours_i = \frac{WBS\ SLOC_i}{Total\ SLOC} * Support\ Hours + WBS\ Hours \qquad (2.1)$$

2.3 Estimate the Effort with Cost Models

The team focused on three software cost estimation models for examination under this project. These were the COCOMO II cost model [2], the C&C CER [4], and the GV CER [4].

2.3.1 COCOMO II

With COCOMO being the most widely used software cost model, this was an easy and obvious choice for the team to evaluate. Only the COCOMO effort formula was used in this evaluation. The COCOMO schedule equation was not used. For ease of comparison, all cost and scale drivers normally used in the COCOMO model were left at nominal values.

$$PM = A * (Size)^E \prod_{t=i}^{n} EM_i \tag{2.2}$$

Where $E = B + 0.01 * \sum_{j=1}^{5} SF_j$
$A = 2.94$, $B = 0.91$
EM: Effort Multipliers
SF: Scale Factors

2.3.2 Command and Control Cost Estimating Relationship

The C&C CER from the "Software Cost Estimation Metrics Manual for Defense Systems" [4] was used to examine the use of the SLOC from the SRDRs, with the team's given assumptions applied to calculated Equivalent Software Lines of Code (ESLOC), in the C&C application type.

$$C\&C\ PM = 6.60 * KESLOC^{1.05} \tag{2.3}$$

2.3.3 Ground Vehicle Cost Estimating Relationship

The GV CER from the "Software Cost Estimation Metrics Manual for Defense Systems" [4] was used to examine the use of the SLOC from the SRDRs, with the team's given assumptions applied to calculated ESLOC, in the GV Operating Environment.

$$GV\ PM = 15.1 * KESLOC^{0.82} \tag{2.4}$$

2.4 Evaluate Estimates Against Actuals

The resulting effort estimates from each model for each of the SRDR data sets was compared against the actual effort reported in the SRDR. The following error and prediction accuracies formulas were used for model comparison.

For each model, for each datapoint i compute:

Percent of Relative Error (RE)

$$RE_i = \frac{actual_i - estimate_i}{actual_i} \tag{2.5}$$

Magnitude Relative Error (MRE)

$$MRE_i = \frac{|actual_i - estimate_i|}{actual_i} \tag{2.6}$$

Statistics for each model across N datapoints in a dataset:

Mean Magnitude Relative Error (MMRE)

$$MMRE = \frac{1}{N}\sum_{i=1}^{N} MRE_i \tag{2.7}$$

Prediction Accuracy (PRED(X)) is the percentage of estimates within X % of the actuals.

The team evaluated the different cost models with respect to 20, 30, and 50 percent prediction accuracy.

$$PRED(X) = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} 1 \text{ if } MRE_i < X \\ 0 \text{ otherwise} \end{cases} \tag{2.8}$$

2.5 Calibrate the Cost Models

The team calibrated the cost models to the entire data set of collected PEO GCS data in order to improve the overall accuracy of the output effort estimations. The full data set was used in order to improve the generic software estimation when a contractor is not known ahead of time. This calibration included data from all 19 projects that were collected for this research. The methods used are given in detail in the COCOMO II Calibration and CER Generation and Calibration sections below.

The team also calibrated the cost models to each contractor individually in order to get contractor specific models. As will be shown in the data, the two contractors had wildly different accuracy using the base cost model predictions, so it made sense to review them separately as different data sets. In this case, the team used the data from the 9 projects from Contractor 1 to calibrate the cost models for Contractor 1, and the team used the 10 data points from Contractor 2 to calibrate the cost models for Contractor 2. The methods used are given in detail in the COCOMO II Calibration and CER Generation and Calibration sections below

2.5.1 COCOMO II Calibration to Local Environment

To calibrate the data for COCOMO II, the team used the methods described in "Software Cost Estimation with COCOMO II" [2] to adjust the A and B coefficients within the COCOMO II equation (Equation 2.2.) The book recommends to calibrate the multiplicative constant (A) if there are at least 5 data points, and calibrate both the multiplicative constant (A), and the baseline exponent (B) if there are at least 10 data points to use. The data sets we are using have 9 data points for contractor 1, 10 data points for contractor 2, and 19 data points for the combined data set. The team decided to calibrate both the multiplicative constant (A), and the baseline exponent (B) for all three data sets even though the first data set had less than 10 data points for three reasons: first, the results from Contractor 1 were so outside the bounds of what was expected that it was decided additional calibration from the original model was needed to get the most accuracy; second, it would be best to compare all three data sets using the same calibration method; third, the other two data sets would also benefit from the additional calibration and met the initial requirements of at least 10 data points.

The equation used from "Software Cost Estimation with COCOMO II" [2] to calibrate both

A and B is shown in Equation 2.9.

$$ln(Effort) - \sum_{n=1}^{5}(\beta_{n+1} * SF_n * ln(Size)) - \sum_{i=1}^{17}(\beta_{i+6} * ln(EM_i)) = \beta_o + \beta_1 * ln(Size) \quad (2.9)$$

where $A = e^{\beta_0}$ and $B = \beta_1$
EM: Effort Multipliers
SF: Scale Factors

The COCOMO II cost model was calibrated using a COCOMO calibration tool from Softstar Systems [18]. The tool is a Microsoft Excel spreadsheet developed by Softstar Systems capable of performing COCOMO II calibrations. The spreadsheet and additional resources, to include a PowerPoint presentation on COCOMO calibrating techniques [19], were downloaded from the Softstar Systems website [18].

The Softstar Systems tool receives inputs in the form of actual effort in Person-Months (P-M), Kilo Software Lines of Code (KSLOC), the product of effort multipliers, and the sum of scale factors for each project. From here, it produces a calibrated multiplier and a calibrated exponent base to use in the COCOMO II effort equation. The calibrated multiplier replaces the A in Equation 2.2, while the calibrated exponent base replaces the B in that same equation.

The team also used instructions contained within the "COCOMO II - Model Definition Manual" [20]

2.5.2 CER Generation and Calibration

To develop a calibrated CER, the team focused on fitting a power equation to the actuals for each contractor. To do this, and also to understand the statistical validity of the models, the team utilized Microsoft Excel scatter plots and power fit, as well as performing a regression analysis on the actual data.

The regression utilizes the natural log of both the effort (response) and the actual P-M of the projects. The team then evaluated the resulting values for statistical values such as F significance, F-value, and P-values to determine statistical relevance and significance.

The team then used the resulting equation for the natural log of the response and took the exponential for each side, creating a $y = A * x^B$ equation. This equation was then used to determine actuals vs. prediction analyses such as MMRE, PRED(20), PRED(30), and PRED(50).This equation was also compared against the calibrated COCOMO II model also performed as part of this study.

2.6 Compare the Cost Models

The team used the relative error equations and prediction accuracies of 20, 30, and 50 percent noted in Section 2.4 for each cost model examined, to include the calibrated models for the combined data and specific contractor data sets, to determine the best cost model for estimating ground vehicle software efforts. In this comparison, the model that presented the least MMRE and greatest PRED(X) would represent the best model to use for future estimation efforts.

CHAPTER 3:
Data Analysis

3.1 Study Background

The data presented in this section is sanitized from identification to protect the proprietary nature of the data.

For this study, the team used data from two separate contractors. SRDR data for three programs at each contractor were reviewed, for a total of six programs reviewed. Among these six programs, nine individual software projects with data were used from the first contractor, and 10 individual software projects with data were used from the second contractor, for a total of 19 individual data points. The organization of these projects is shown in Table 3.1.

Table 3.1. Contractor Program Efforts

Contractor 1		Contractor 2	
Program 1	Project A	Program 4	Project J
	Project B		Project K
	Project C	Program 5	Project L
Program 2	Project D		Project M
	Project E	Program 6	Project N
Program 3	Project F		Project O
	Project G		Project P
	Project H		Project Q
	Project I		Project R
			Project S

The team analyzed the above data using COCOMO II from "Software Cost Estimation with COCOMO II" [2], the C&C CER, and the GV CER from "The Software Cost Estimation Metrics Manual for Defense Systems" [4]. For COCOMO II, nominal scale factors were assumed for all data inputs due to insufficient data to determine otherwise, SLOC data was adjusted based on the SRDR data, and the estimated P-M were calculated using Equation 2.2 and standard COCOMO II.2000 constants.

For both of the CERs, the team used the known SLOC values with the given equations. Equation 2.3 was used for the C&C CER P-M, and Equation 2.4 was used for the GV CER P-M.

The resulting values were compared with the actual hours used on each of these programs, to see if there was a strong correlation between any of these CERs and the actual cost of each program. The actual hours were then updated to include all software support and development hours that were not directly related to the SLOC production by giving each effort a weighted average of those hours based on the percentage of ESLOC for that effort over the entire ESLOC for that contract.

The data used in this study is proprietary Controlled Unclassified Information (CUI), requiring the data to be sterilized by removing any identifying information. All of the data was for similar types of systems, that could be considered Ground Vehicles based on the definitions defined in "The Software Cost Estimation Metrics Manual for Defense Systems" [4]. Each system was also labeled as a Command and Control system within the SRDR, which is why we used both CERs in our analysis.

3.2 Study Checklist

Checklists are a standard tool and common practice among current professionals. Magne Jørgensen developed "A Preliminary Checklist for Software Cost Management" [21] that outlines a preliminary process and framework, as well as a checklist for software cost management. Robert Park developed a number of standard checklists that guide both the development of software cost estimates and the management and evaluation of cost estimates. "A Manager's Checklist for Validating Software Cost and Schedule Estimates" [22], from Robert Park, "provides a checklist of questions to ask and evidence to look for when assessing the credibility of a software cost and schedule estimate" [22] and would be most useful when evaluation and offeror's estimate. "Checklists and Criteria for Evaluating the Cost and Schedule Estimating Capabilities of Software Organizations" [23], also from

Robert Park, provides checklists and guidance for creating a software cost estimate, which would be useful for any organization looking to estimate a project.

The team developed a checklist throughout the conduct of this capstone, tailored to this effort, to help guide the process, facilitate making the study repeatable, and increase communicability of the study. These standard checklists already developed are designed to evaluate the maturity of an offeror's estimation process [23], evaluate the quality of the support that an organization provides for the estimation process [23], and to assess the credibility of software cost and schedule estimates [22]. The goal of any checklist is to guide the process and force critical thinking, to ensure important steps are not missed and all information is taken into account.

The team's checklist is below:

- Collect project effort and size data (SRDR is a good reference for this information if available)
  - Evaluate project productivity for reasonableness
- Implement assumptions and calculate project size (ESLOC)
- Estimate the effort using a preferred cost model
- Evaluate the estimates against actuals
- Calibrate the cost model using the preferred hierarchy of data sources below
  - Historical project data
  - Data from the same contractor on similar efforts
  - Data from similar domains or programs from other contractors
- Verify calibrated model

The team used this checklist to guide each run through a software cost estimating model. It maintained consistency within the study and allowed for the process to be repeatable and more easily communicated during the conduct of this capstone.

3.3 Implementing Assumptions and Calculating ESLOC

The data was used to calculate an ESLOC creation for each effort based on an estimate of 100% of all new code, 50% of modified existing code, 20% of unmodified existing code, and 50% of deleted code. This estimate was in accordance with the method that Contractor 1 used to estimate their ESLOC in their WBS dictionary.

$$\begin{aligned} ESLOC = New\,Code * 100\% + Modified\,Code * 50\% \\ +Unmodified\,Reused\,Code * 20\% + Deleted\,Code * 50\% \end{aligned} \tag{3.1}$$

Table 3.2 and Table 3.3 show the SLOC breakdown found in the contractor SRDRs and ESLOC as calculated using Equation 3.1.

Table 3.2. Contractor 1 ESLOC

Project	New Code	Modified	Re-use	Deleted	ESLOC
A	219956	23004	401715	242671	433136.5
B	20014	8584	77887	6605	43185.9
C	213058	6133	6133	5211	219956.45
D	35173	15602	876466	30943	233738.7
E	7125	27795	138515	3981	50716
F	1154	3642	31350	548	9519
G	5687	1134	328966	548	72321.2
H	171467	25265	252623	18298	243773.1
I	286	1432	4014	23922	13765.8

Data sanitized to protect proprietary data.

Table 3.3. Contractor 2 ESLOC

Project	New Code	Modified	Re-use	Deleted	ESLOC
J	140809	0	170078	0	174825
K	24027	0	2679	0	24563
L	163377	104	2973	0	164024
M	61908	17186	257488	0	121999
N	2204	782	15378	0	5671
O	5708	2567	23103	0	11612
P	42293	2475	21392	0	47809
Q	27480	24040	41450	0	47790
R	17068	3378	243236	0	67404
S	2094	469	52715	0	12872

Data sanitized to protect proprietary data.

Tables 3.4 and 3.5 show the reported hours from the contractor SRDRs and adjusted hours and P-M using Equation 2.1.

Table 3.4. Contractor 1 Effort

Project	Hours	Adjusted Hours	Effort (P-M)	Adjusted Effort (P-M)
A	24952	45199.83	155.95	282.50
B	23058	25076.81	144.11	156.73
C	13678	23960.30	85.49	149.75
D	8671	8671.00	52.55	52.55
E	3023.2	3023.20	18.32	18.32
F	442.3	447.95	2.68	2.71
G	896	938.95	5.43	5.69
H	42877	43021.76	259.86	260.74
I	542	550.17	3.28	3.33

Data sanitized to protect proprietary data.

Table 3.5. Contractor 2 Effort

Project	Hours	Adjusted Hours	Effort (P-M)	Adjusted Effort (P-M)
J	199125.79	211746.00	1244.54	1323.41
K	2058.90	3832.04	12.87	23.95
L	53869.00	53869.00	336.68	336.68
M	39729.00	39729.00	254.67	254.67
N	9676.33	9809.85	60.48	61.31
O	10873.88	11147.30	67.96	69.67
P	15115.63	16241.36	94.47	101.51
Q	16758.00	17883.28	104.74	111.77
R	22151.56	23738.68	138.45	148.37
S	34303.00	34606.08	214.39	216.29

Data sanitized to protect proprietary data.

3.3.1 Data Assumptions

Program 1 used 160 hours per P-M based on the data given from the SRDR WBS dictionary. Program 2 used 165 hours per P-M based on the data given from the SRDR WBS dictionary. Program 3 used an average of 165 hours per P-M according to the SRDR WBS dictionary. Program 4 used 160 hours per P-M based on the data given from the SRDR WBS dictionary. Program 5 used 160 hours per P-M for Project L and 156 hours per P-M for Project M based on information given by the SRDR WBS Dictionary. Program 6 assumed $100/hr for software data that had only cost and SLOC data and no hours. Program 6 assumed 160 hours per P-M, as that data was not provided, and that value was used in other contracts from the same contractor. The Effort Multipliers (EMs) were assumed to be nominal for the data, which set the EM = 1. The Scale Factors (SFs) were assumed to be nominal for the data, which set E = 1.0997.

3.3.2 Productivity

Initial review of the data indicates Contractor 1 and Contractor 2 have far different productivity when it comes to ESLOC/hour. The estimates for Contractor 1 are orders of magnitude higher than the actuals, while they are much closer in Contractor 2. This is shown in Table 3.6.

Table 3.6. Productivity

Project	Hours	Adjusted Hours	ESLOC	Productivity	Adjusted Productivity
A	24952	45199.83	433136.5	2777.41	1533.23
B	23058	25076.81	43185.9	299.67	275.54
C	13678	23960.30	219956.45	2572.97	1468.81
D	8671	8671.00	233738.7	4447.80	4447.80
E	3023.2	3023.20	50716	2767.97	2767.97
F	442.3	447.95	9519	3551.06	3506.25
G	896	938.95	72321.2	13318.08	12708.93
H	42877	43021.76	243773.1	938.09	934.94
I	542	550.17	13765.8	4190.70	4128.43
J	199125.79	211746.00	174824.6	140.47	132.10
K	2058.9	3832.04	24562.8	1908.81	1025.58
L	53869	53869.00	164023.6	487.18	487.18
M	39729	39729.00	121998.6	479.04	479.04
N	9676.33	9809.85	5670.6	93.76	92.49
O	10873.88	11147.30	11612.1	170.86	166.67
P	15115.63	16241.36	47808.9	506.06	470.98
Q	16758	17883.28	47790	456.28	427.57
R	22151.56	23738.68	67404.2	486.86	454.31
S	34303	34606.08	12871.5	60.04	59.51

Contractor 1 completed projects A thru I. Contractor 2 completed projects J thru S.

3.4 Data Anomalies and Irregularities

Both contractors' reported data include high average productivity required to perform the ESLOC calculated by their SRDR data. Initially, it was conjectured that this may be due to reporting computer generated code as new developed code or in assumptions made by the contractor in their Adaptation Adjustment Factor (AAF). Because of this, the team believed that this may be able to be addressed by calibration of the data.

However, Project G from Contractor 1 has reported an exceptionally large amount of ESLOC, yet a small amount of actual man-hours to perform this effort. Calibrating of the data still resulted in that data point being an outlier. The required productivity to complete a project of this size with such little actual effort is 12708 ESLOC/P-M, as shown in Table 3.6. This value far exceeds all industry benchmarks as shown in Table 3.7 which can only be explained by one of the following: the contractor's personnel are far more productive than industry standards, or there are errors in the SRDR data reporting (to include SLOC, AAF methodology, types of SLOC, and/or actual man-hours). Assuming the hours reported are accurate, the most likely candidate is errors in the SLOC measurement or reporting. Coupling this with Table 3.3 which shows this data point as an outlier lends credence to this data being spurious in nature.

Figure 3.1 provides definitions for each statistic in Table 3.7

Table 3.7. Productivity Benchmarks

OpEnv / AppType	N	LCI	Mean	UCI	SE	SD	Q1	Mdn.	Q3
GV	26	110.2	140.6	171	14.8	75.3	89.9	121	169.5
C&C	33	120.5	140.7	160.8	9.9	56.8	103.1	128.2	177.8

Source: Software Cost Estimation Metrics Manual for Defense Systems. Productivity Benchmarks

Statistic	Description
N	Number of data points
Min ESLOC	Minimum size in thousands of equivalent source lines of code
Max ESLOC	Maximum size in thousands of equivalent source lines of code
LCI	Lower Confidence Interval is an estimate of an interval below the sample mean within which the population mean is estimated to lie
Mean	Estimated sample value representing the population central value; equal to the sum of the values divided by the number of values, i.e., arithmetic mean
UCI	Upper Confidence Interval is an estimate of an interval above the sample mean within which the population mean is estimated to lie
SE	Standard Error of the mean is an estimate computed from the sample of data being analyzed of the standard deviation of sample means over all possible samples (of a given size) drawn from the population. It is found by dividing the StdDev by the square root of N.
StdDev	Standard Deviation measures the amount of variation or dispersion from the mean in a sample. Plus or minus one standard deviation from the mean is a range that includes about 68% of the data.
Q1	Numerical value for the lower 25% of ranked data (1st Quartile), i.e., the value half way between the lowest value and the median in a set of ranked values
Median	Numerical value separating the higher half of a sample from the lower half, i.e., the middle value in a set of ranked values
Q3	Numerical value for the lower 75% of ranked data (3rd Quartile), i.e. the value half way between the median and the highest value in a set of ranked values

Figure 3.1. Productivity Benchmark Statistic Definitions. Source: [4]

Figure 3.2 shows a sampling of four of Contractor 1's data points and shows that three are grossly outside of the expected values when plotted with the 300-plus data points used in generating the CERs found in the Software Cost Estimation Metrics Manual for Defense Systems [4].

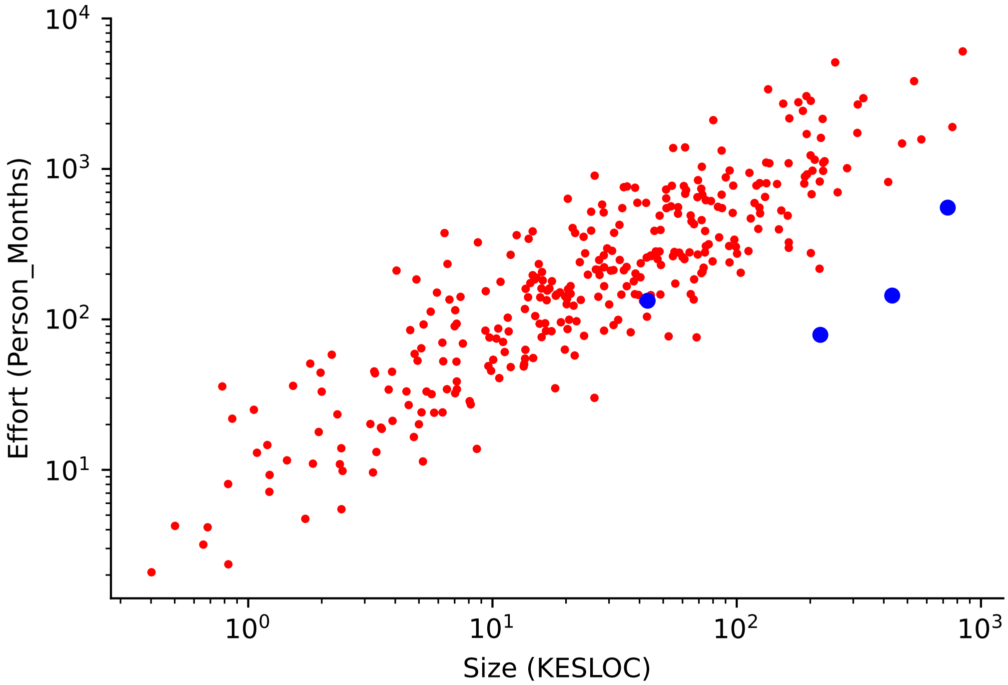

Figure 3.2. Study Data Plotted Against Historical DoD Data. Source: [24], Adapted from [4]

[1]

[1]Figure 3.2 was obtained from personal correspondence with Dr. Raymond Madachy.

3.5 Cost Model Estimations

Tables 3.8 and 3.9 show the estimated values for the effort based on the COCOMO II, C&C CER, and GV CER.

Table 3.8. Contractor 1 Model Estimates

Project	COCOMO II (P-M)	C&C CER (P-M)	GV CER (P-M)
A	2332.62	3872.58	2192.85
B	184.81	344.07	331.10
C	1107.17	1901.07	1258.04
D	1183.70	2026.33	1322.32
E	220.54	407.33	377.74
F	35.04	70.32	95.81
G	325.82	591.25	505.33
H	1239.70	2117.77	1368.69
I	52.56	103.58	129.66

Table 3.9. Contractor 2 Model Estimates

Project	COCOMO II (P-M)	C&C CER (P-M)	GV CER (P-M)
J	860.08	1493.75	1042.10
K	99.37	190.25	208.45
L	801.83	1397.00	989.01
M	579.04	1023.80	775.87
N	19.82	40.82	62.65
O	43.59	86.64	112.77
P	206.68	382.85	359.89
Q	206.59	382.69	359.78
R	301.55	549.12	476.98
S	48.82	96.53	122.71

The graph in Figure 3.3 shows the relative error for each cost model broken out by contractor. This graph shows that the variance is much higher for Contractor 1 than Contractor 2, but in both cases COCOMO II has the smallest maximum relative error.

For the following data, KTR represents Contractor where space is limited.

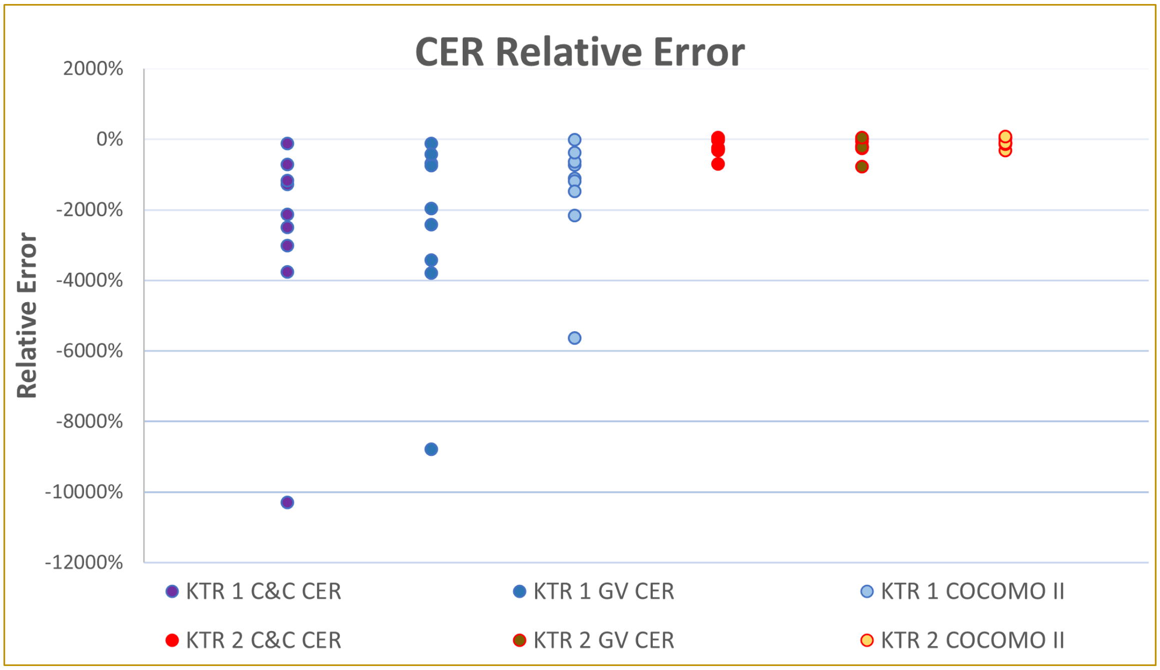

Figure 3.3. Base Cost Model Relative Error

The scatter plot in Figure 3.4 shows the base cost model results with respect to the actual P-M spent on each program. The perfect fit line is shown to illustrate what the data would look like if the CER data perfectly modeled the effort required. The distance from this line indicates the relative error of each data point.

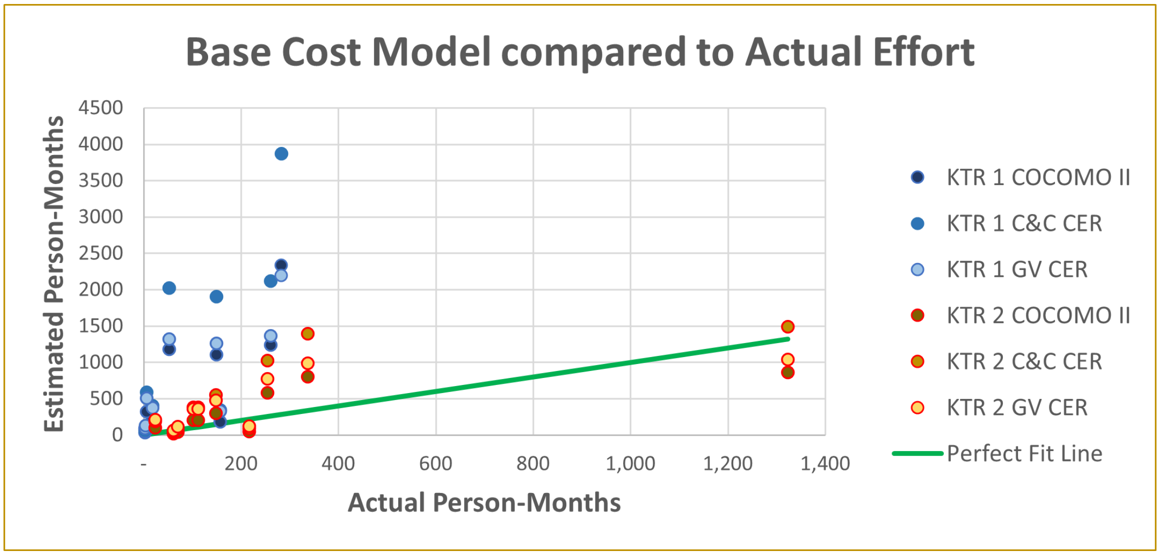

Figure 3.4. Scatter Plot of CER Results Compared to Actual P-M

3.6 Calibrating the Cost Models

Based on the above analysis, the team calibrated each of the cost models to each unique environment to explore the effects. This was accomplished using both the data as a complete set and calibrating each contractor separately using their respective data. Each of these sets of data were calibrated by using the methods described in Section 2.5 of this report.

3.6.1 COCOMO II Calibration

The COCOMO II model was calibrated using the method described in Section 2.5.1. Calibrating COCOMO II using all combined data resulted in Equation 3.2 below.

$$Person\ Months = A * (Size)^E \prod_{t=i}^{n} EM_i \tag{3.2}$$

Where E = B + 0.01* $\sum_{j=1}^{5} SF_j$

A = 4.559, B = 0.51

EM: Effort Multipliers = 1

SF: Scale Factors = 18.97

Calibrating COCOMO II using just Contractor 1 data resulted in Equation 3.3 below.

$$Person\ Months = A * (Size)^E \prod_{t=i}^{n} EM_i \tag{3.3}$$

where E = B + 0.01*$\sum_{j=1}^{5} SF_j$

A = 0.2236, B = 0.97

EM: Effort Multipliers = 1

SF: Scale Factors = 18.97

Calibrating COCOMO II using just Contractor 2 data resulted in Equation 3.4 below.

$$Person\ Months = A * (Size)^E \prod_{t=i}^{n} EM_i \tag{3.4}$$

where E = B + 0.01*$\sum_{j=1}^{5} SF_j$

A = 13.99, B = 0.44

EM: Effort Multipliers = 1

SF: Scale Factors = 18.97

3.6.2 CER Calibration

The actual effort and Kilo Equivalent Software Lines of Code (KESLOC) for the data sets: Combined data, Contractor 1, and Contractor 2 were each arranged into Microsoft Excel for regression analysis. A natural log of the response value and independent variable were taken to support a linear regression which then was converted back into an power relationship. A linear regression analysis between these variables were conducted for each, resulting in the statistical outcomes outlined in Tables 3.10, 3.11, and 3.12. The significance F values for all three models (Combined Data, Contractor 1, and Contractor 2) were all found to be less than 0.05, coupled with high F-values in comparison meaning that there is a low probability that the resulting equations are not valid.

Table 3.10. Combined Data Regression Analysis

Regression Statistics					
Multiple R	0.537458				
R Square	0.288862				
Adjusted R Square	0.24703				
Standard Error	1.431346				
Observations	19				
	Coefficients	Standard Error	t Stat	P-value	
Intercept	1.517695	1.111956644	1.364887	0.190082	
X Variable 1	0.695911	0.264826634	2.6278	0.017635	
	df	SS	MS	F	Significance F
Regression	1	14.14729856	14.1473	6.905332	0.017634936
Residual	17	34.8287492	2.04875		
Total	18	48.97604776			

The Combined data regression analysis completed with a low significance F and high F, which means that the probability of the regression analysis results being incorrect is low. The t stat and P-values for the x variable coefficient are both acceptable, leading to the conclusion that this coefficient is statistically significant to the response. The team decided to include the intercept even with a high p-value, as it is used for scaling in the effort model. Converting from this natural log back into an exponential relationship resulted in a CER represented by Equation 3.5.

$$Person\ Months = 4.5617 * \mathrm{KESLOC}^{0.6959} \tag{3.5}$$

This is also shown in the following graph, Figure 3.5, depicting the calibrated CER and actuals for the combined data. It is noted that this CER results in a very low R^2 number, indicating that the model does not explain the response data well. However, for further model analyses the MMRE, PRED(20), PRED(30) and PRED(50) values will be used to indicate the quality of the model.

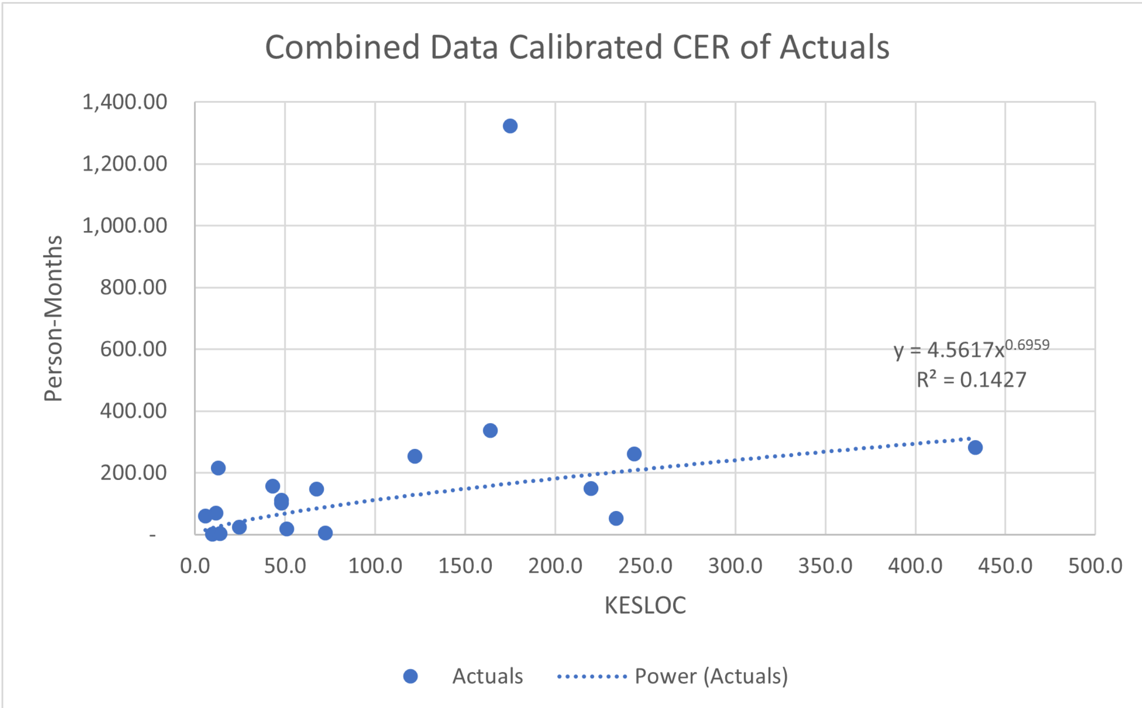

Figure 3.5. Combined Data Calibrated CER and Actuals

The team then applied the same process to the Contractor 1 data, resulting in the statistical outcomes outlined in Table 3.11.

Table 3.11. Contractor 1 Regression Analysis

Regression Statistics					
Multiple R	0.824114				
R Square	0.679164				
Adjusted R Square	0.63333				
Standard Error	1.146946				
Observations	9				
	Coefficients	Standard Error	t Stat	P-value	
Intercept	-1.49526	1.368948145	-1.09227	0.310876	
X Variable 1	1.159857	0.301307216	3.849416	0.006297	
	df	SS	MS	F	Significance F
Regression	1	19.49284831	19.49285	14.818	0.006297208
Residual	7	9.208390109	1.315484		
Total	8	28.70123842			

The Contractor 1 regression analysis completed with a low significance F and high F, which means that the probability of the regression analysis results being incorrect is low. The t stat and P-values for the X variable is acceptable, leading to the conclusion that this coefficient is statistically significant to the response. The intercept is showing a high P-value, which indicates possible statistical significance issues with this variable, but the team has decided to include this in the model as it is used for scaling the effort. Converting from this natural log back into a power relationship resulted in a CER represented by Equation 3.6.

$$Person\ Months = 0.2242 * \text{KESLOC}^{1.1599} \tag{3.6}$$

This is also shown in the following graph, Figure 3.6, depicting the calibrated CER and actuals for the Contractor 1 data. The R^2 value of this regression model is 0.6236, indicating that 62.36% of the variability in the response is explained by this model. As one can see from the chart, there is significant dispersion from the predicted values of this model, further analyses regarding the accuracy of the model is warranted using the MMRE and PRED(X) values as an indicator of model quality.

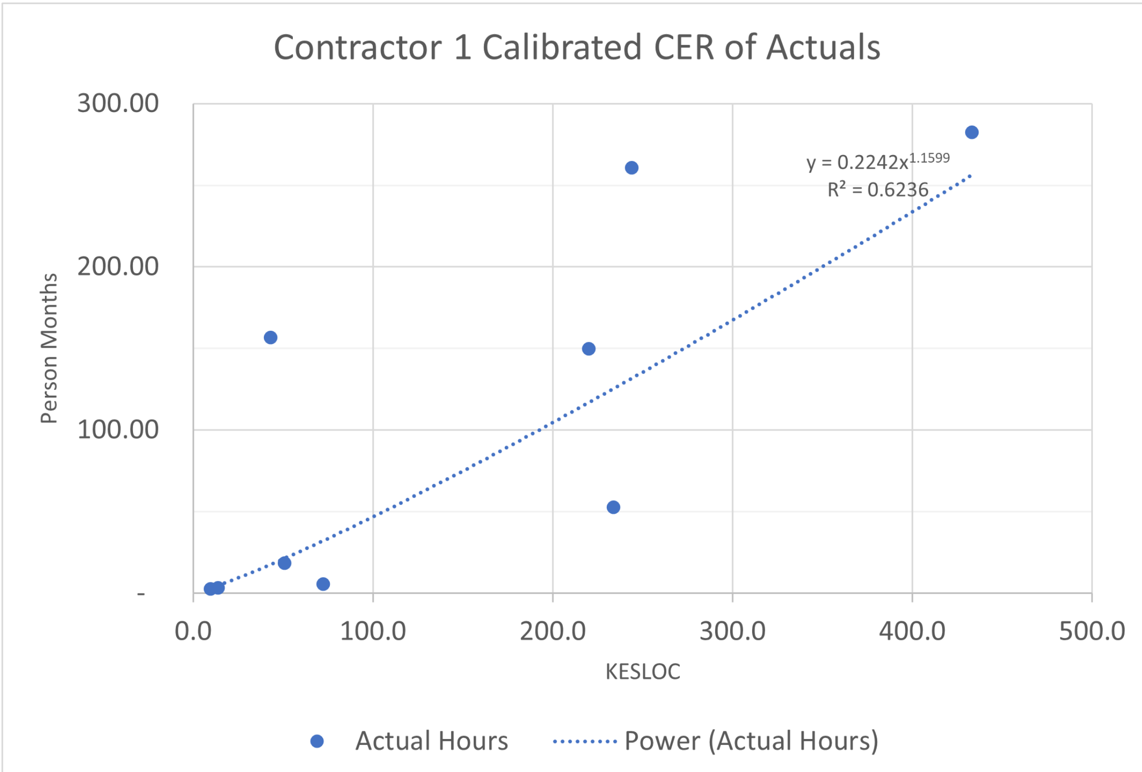

Figure 3.6. Contractor 1 Calibrated CER and Actuals

Finally, the team conducted the same process for Contractor 2 data, resulting in the following outcomes outlined in Table 3.12.

Table 3.12. Contractor 2 Regression Analysis

Regression Statistics					
Multiple R	0.68501				
R Square	0.469238				
Adjusted R Square	0.402893				
Standard Error	0.846188				
Observations	10				
	Coefficients	Standard Error	t Stat	P-value	
Intercept	2.638042	0.920220151	2.866751	0.020933	
X Variable 1	0.633597	0.238243386	2.659451	0.02883	
	df	SS	MS	F	Significance F
Regression	1	5.064286857	5.064287	7.072681	0.028830005
Residual	8	5.728279753	0.716035		
Total	9	10.79256661			

The Contractor 2 regression analysis completed with a low significance F and high F, which means that the probability of the regression analysis results being incorrect is low. The t stat and P-values for the coefficients are both acceptable, leading to the conclusion that the coefficients are statistically significant to the response. Converting from this natural log back into an power relationship resulted in a CER represented by equation 3.7.

$$Person\ Months = 13.985 * \text{KESLOC}^{0.6336x} \tag{3.7}$$

This is also shown in the following graph, Figure 3.7, depicting the calibrated CER and actuals for the Contractor 2 data. The R^2 value of this regression model is 0.4815 indicating that 48.15% of the variability in the response is explained by this model. In looking at the data this is highly influenced by a single data point and may not be reflective of the fit if that data point is determined to be an outlier, as the fit to all other data is noticeably better in the graph. MMRE and PRED(X) values will be used to indicate the quality of the cost estimation models.

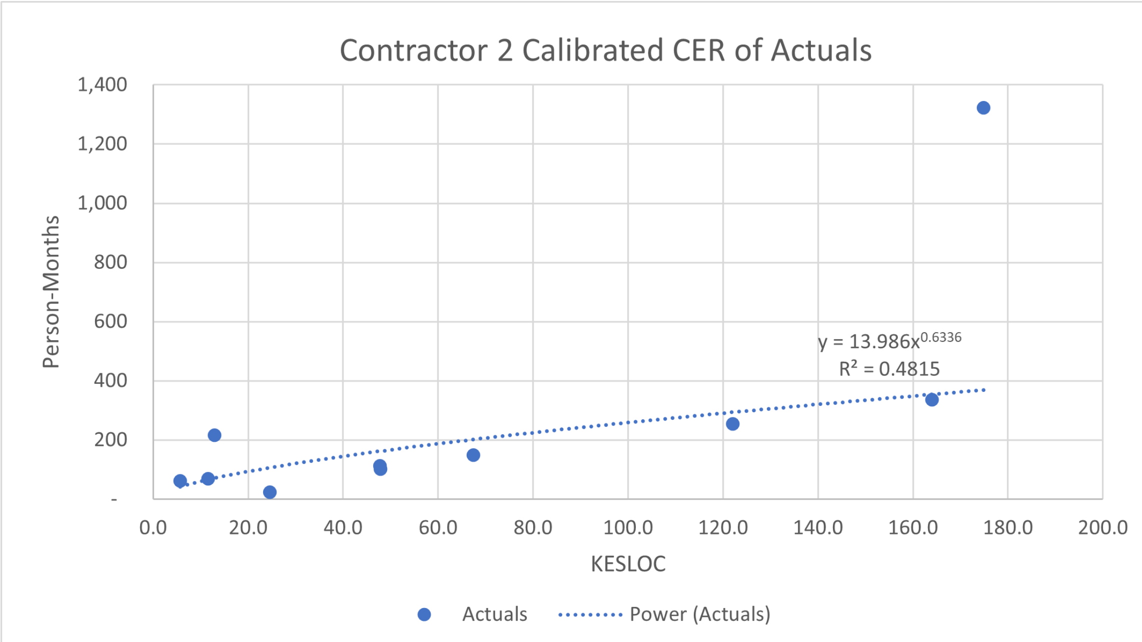

Figure 3.7. Contractor 2 Calibrated CER and Actuals

3.7 Evaluating and Comparing the Cost Models

This section will cover the results of the cost models and compare them to each other by measuring their relative error and prediction accuracy. The section will be broken into the base results, combined calibrated results, and contractor specific calibrated results.

3.7.1 Base Cost Model Results

Table 3.13 shows the actual effort data, the estimated effort, the relative error, and the MRE for the uncalibrated COCOMO II model for each project and contractor. It also shows the prediction accuracy of the data for 20%, 30%, and 50% magnitudes.

Table 3.13. COCOMO II Model Evaluation of Combined Data

Project	Actual Effort (P-M)	Estimated Effort (P-M)	Relative Error	Magnitude Relative Error
A	282.50	2332.62	-725.7%	725.7%
B	156.73	184.81	-17.9%	17.9%
C	149.75	1107.17	-639.3%	639.3%
D	52.55	1183.70	-2152.4%	2152.4%
E	18.32	220.54	-1103.7%	1103.7%
F	2.71	35.04	-1190.5%	1190.5%
G	5.69	325.82	-5625.7%	5625.7%
H	260.74	1239.70	-375.5%	375.5%
I	3.33	52.56	-1476.4%	1476.4%
J	1323.41	860.08	35.0%	35.0%
K	23.95	99.37	-314.9%	314.9%
L	336.68	801.83	-138.2%	138.2%
M	254.67	579.04	-127.4%	127.4%
N	61.31	19.82	67.7%	67.7%
O	69.67	43.59	37.4%	37.4%
P	101.51	206.68	-103.6%	103.6%
Q	111.77	206.59	-84.8%	84.8%
R	148.37	301.55	-103.2%	103.2%
S	216.29	48.82	77.4%	77.4%
			MMRE	757.7%
			PRED(20)	5.3%
			PRED(30)	5.3%
			PRED(50)	15.8%

Table 3.14 shows the actual effort data, the estimated effort, the relative error, and the MRE for the C&C CER for each effort and contractor. It also shows the prediction accuracy of the data for 20%, 30%, and 50% magnitudes.

Table 3.14. C&C CER Model Evaluation of Combined Data

Project	Actual Effort (P-M)	Estimated Effort (P-M)	Relative Error	Magnitude Relative Error
A	282.50	3872.58	-1270.8%	1270.8%
B	156.73	344.07	-119.5%	119.5%
C	149.75	1901.07	-1169.5%	1169.5%
D	52.55	2026.33	-3755.9%	3755.9%
E	18.32	407.33	-2123.1%	2123.1%
F	2.71	70.32	-2490.1%	2490.1%
G	5.69	591.25	-10290.0%	10290.0%
H	260.74	2117.77	-712.2%	712.2%
I	3.33	103.58	-3006.5%	3006.5%
J	1323.41	1493.75	-12.9%	12.9%
K	23.95	190.25	-694.4%	694.4%
L	336.68	1397.00	-314.9%	314.9%
M	254.67	1023.80	-302.0%	302.0%
N	61.31	40.82	33.4%	33.4%
O	69.67	86.64	-24.4%	24.4%
P	101.51	382.85	-277.2%	277.2%
Q	111.77	382.69	-242.4%	242.4%
R	148.37	549.12	-270.1%	270.1%
S	216.29	96.53	55.4%	55.4%
			MMRE	1429.7%
			PRED(20)	5.3%
			PRED(30)	10.5%
			PRED(50)	15.8%

Table 3.15 shows the actual effort data, the estimated effort, the relative error, and the MRE for the GV CER for each effort and contractor. It also shows the prediction accuracy of the data for 20%, 30%, and 50% magnitudes.

Table 3.15. GV CER Model Evaluation of Combined Data

Project	Actual Effort (P-M)	Estimated Effort (P-M)	Relative Error	Magnitude Relative Error
A	282.50	2192.85	-676.2%	676.2%
B	156.73	331.10	-111.3%	111.3%
C	149.75	1258.04	-740.1%	740.1%
D	52.55	1322.32	-2416.2%	2416.2%
E	18.32	377.74	-1961.6%	1961.6%
F	2.71	95.81	-3429.2%	3429.2%
G	5.69	505.33	-8780.1%	8780.1%
H	260.74	1368.69	-424.9%	424.9%
I	3.33	129.66	-3788.5%	3788.5%
J	1323.41	1042.10	21.3%	21.3%
K	23.95	208.45	-770.4%	770.4%
L	336.68	989.01	-193.8%	193.8%
M	254.67	775.87	-204.7%	204.7%
N	61.31	62.65	-2.2%	2.2%
O	69.67	112.77	-61.9%	61.9%
P	101.51	359.89	-254.5%	254.5%
Q	111.77	359.78	-221.9%	221.9%
R	148.37	476.98	-221.5%	221.5%
S	216.29	122.71	43.3%	43.3%
			MMRE	1280.2%
			PRED(20)	5.3%
			PRED(30)	10.5%
			PRED(50)	15.8%

The relative error describes how far from the actual data each estimate is, which gives an indicator of accuracy for each of the estimates. The tables also display the prediction accuracy or PRED(X) of each model to show what percentage of each CER is within 20%, 30%, or 50% respectively (for PRED(20), PRED(30), or PRED(50)).

Figure 3.8 shows the Relative Error of each cost model graphically to show the differences between the different contractors and methods. It is shown visually that the Contractor 1 estimates are far less accurate the lower the actual effort was, with some relative error data points multiple orders of magnitude from the actual effort.

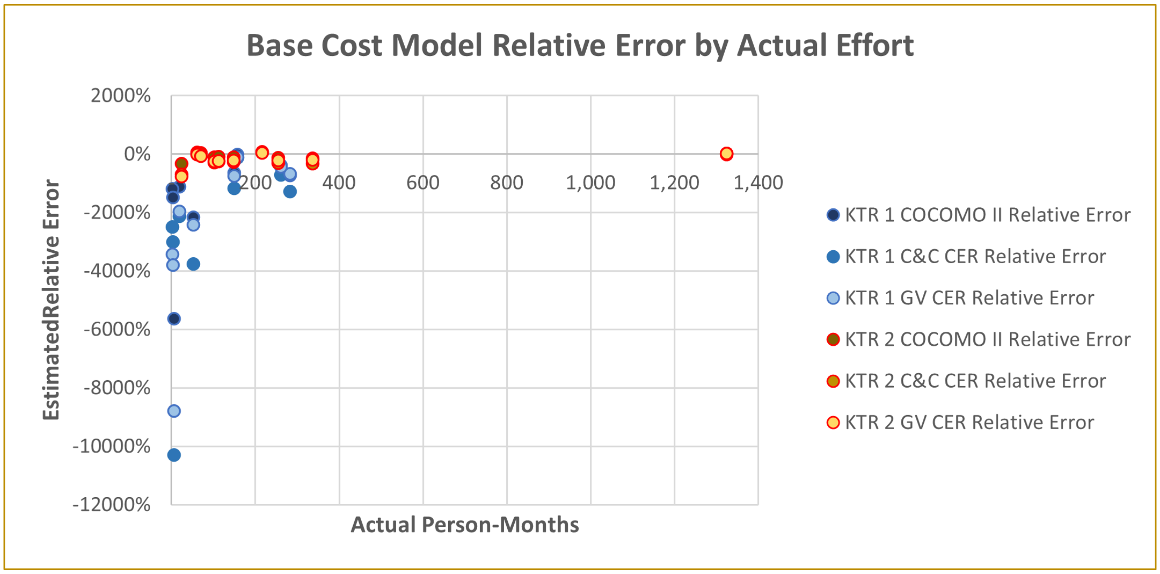

Figure 3.8. Scatter Plot of CER Relative Errors Compared to Actual P-M

The data in these tables indicates that none of the models are very close to the actuals without calibration (with the best model having only 10% of the data points within 30% of the actual data when you account for both contractors, or only 20% of the data points when you break out the data by contractor). The data also indicates that the models are highly contractor dependent, with massively different MMREs depending on which contractor is being measured.

3.7.2 Combined Calibrated Cost Model Results

From here, the calibrated results will be shown, starting with the combined calibration including all data points from the study. Table 3.16 shows the calibrated COCOMO II data, with MMRE and PRED(X) results. Table 3.17 shows the calibrated CER data, with MMRE and PRED(X) results.

Table 3.16. Calibrated COCOMO II Model Evaluation of Combined Data

Project	Actual Effort (P-M)	Estimated Effort (P-M)	Relative Error	Magnitude Relative Error
A	282.50	255.37	9.6%	9.6%
B	156.73	17.62	88.8%	88.8%
C	149.75	116.38	22.3%	22.3%
D	52.55	124.88	-137.6%	137.6%
E	18.32	21.23	-15.9%	15.9%
F	2.71	3.05	-12.4%	12.4%
G	5.69	32.04	-463.0%	463.0%
H	260.74	131.12	49.7%	49.7%
I	3.33	4.68	-40.3%	40.3%
J	1323.41	165.87	87.5%	87.5%
K	23.95	42.32	-76.7%	76.7%
L	336.68	158.67	52.9%	52.9%
M	254.67	129.12	49.3%	49.3%
N	61.31	15.25	75.1%	75.1%
O	69.67	25.12	63.9%	63.9%
P	101.51	67.27	33.7%	33.7%
Q	111.77	67.25	39.8%	39.8%
R	148.37	85.44	42.4%	42.4%
S	216.29	26.99	87.5%	87.5%
			MMRE	76.2%
			PRED(20)	15.8%
			PRED(30)	21.1%
			PRED(50)	52.6%

Table 3.17. Calibrated CER Model Evaluation of Combined Data

Project	Actual Effort (P-M)	Estimated Effort (P-M)	Relative Error	Magnitude Relative Error
A	282.50	308.80	-9.3%	9.3%
B	156.73	62.07	60.4%	60.4%
C	149.75	192.70	-28.7%	28.7%
D	52.55	201.02	-282.5%	282.5%
E	18.32	69.42	-278.9%	278.9%
F	2.71	21.67	-698.2%	698.2%
G	5.69	88.86	-1461.6%	1461.6%
H	260.74	206.99	20.6%	20.6%
I	3.33	28.01	-740.1%	740.1%
J	1323.41	164.24	87.6%	87.6%
K	23.95	41.91	-75.0%	75.0%
L	336.68	157.11	53.3%	53.3%
M	254.67	127.86	49.8%	49.8%
N	61.31	15.11	75.4%	75.4%
O	69.67	24.88	64.3%	64.3%
P	101.51	66.62	34.4%	34.4%
Q	111.77	66.60	40.4%	40.4%
R	148.37	84.61	43.0%	43.0%
S	216.29	26.73	87.6%	87.6%
			MMRE	220.6%
			PRED(20)	5.3%
			PRED(30)	15.8%
			PRED(50)	36.8%

Figure 3.9 shows the Relative Error of both the original base cost models, and the contractor calibrated cost models graphically to show the improvements made from calibration. We can see graphically that calibration reduced the variation quite a bit for both models.

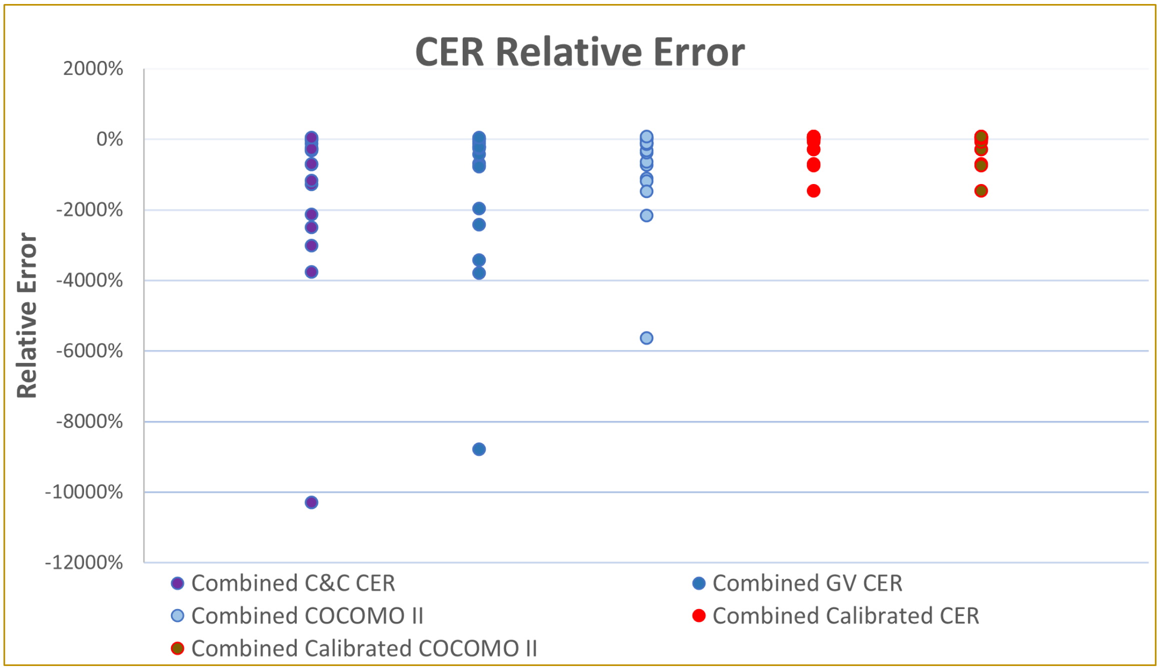

Figure 3.9. Relative Error of Base and Calibrated Combined Cost Models P-M

Figure 3.10 shows the Relative Error of each cost model graphically to show the differences between the different contractors and methods. As with the uncalibrated data, the data with low actual P-M are shown to have the most relative error.

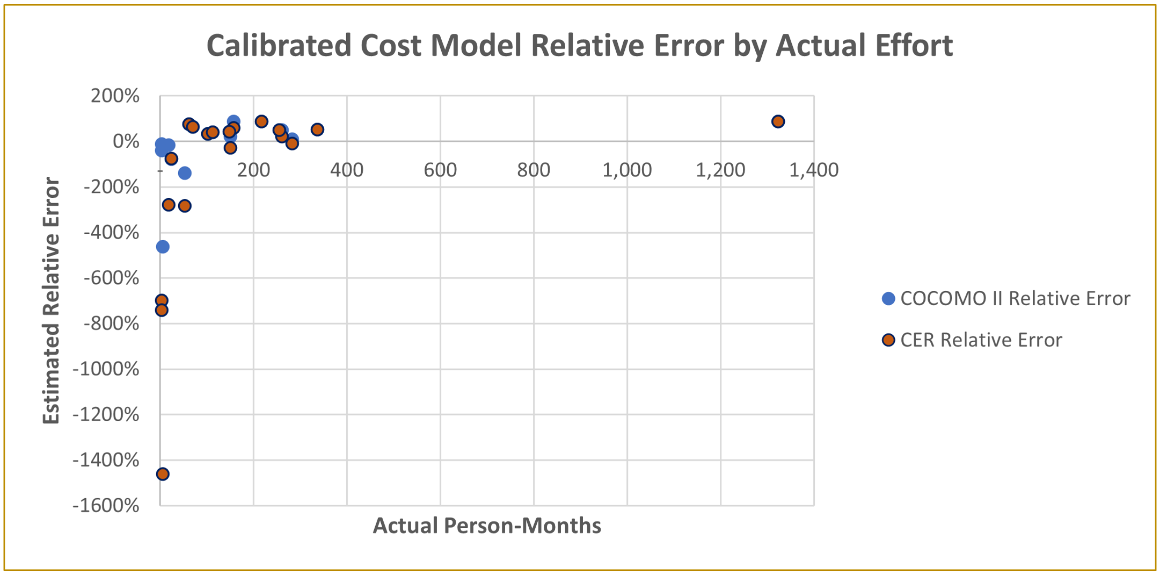

Figure 3.10. Scatter Plot of Cost Model Relative Errors Compared to Actual P-M

The data shows that both MMRE and PRED(30) values improved with this combined calibration for both COCOMO II and the CER results. Between the two cost models, the combined calibrated COCOMO II data had the best MMRE and Prediction Accuracy.

3.7.3 Contractor Calibrated Cost Model Results

The contractor specific data was used to calibrate both the COCOMO II cost model, and the CERs using the methods described in Section 2.5. The results for each contractor is shown below.

Table 3.18 shows the contractor calibrated COCOMO II data for contractor 1, with MMRE and PRED(X) results.

Table 3.18. COCOMO II Calibrated by Contractor 1 Evaluation

Project	Actual Effort (P-M)	Estimated Effort (P-M)	Relative Error	Magnitude Relative Error
A	282.50	255.37	9.6%	9.6%
B	156.73	17.62	88.8%	88.8%
C	149.75	116.38	22.3%	22.3%
D	52.55	124.88	-137.6%	137.6%
E	18.32	21.23	-15.9%	15.9%
F	2.71	3.05	-12.4%	12.4%
G	5.69	32.04	-463.0%	463.0%
H	260.74	131.12	49.7%	49.7%
I	3.33	4.68	-40.3%	40.3%
			MMRE	93.3%
			PRED(20)	33.3%
			PRED(30)	44.4%
			PRED(50)	66.7%

Table 3.19 shows the contractor calibrated COCOMO II data for Contractor 2, with MMRE and PRED(X) results.

Table 3.19. COCOMO II Calibrated by Contractor 2 Evaluation

Project	Actual Effort (P-M)	Estimated Effort (P-M)	Relative Error	Magnitude Relative Error
J	1323.41	361.33	72.7%	72.7%
K	23.95	105.00	-338.4%	338.4%
L	336.68	347.11	-3.1%	3.1%
M	254.67	288.09	-13.1%	13.1%
N	61.31	41.72	32.0%	32.0%
O	69.67	65.51	6.0%	6.0%
P	101.51	159.71	-57.3%	57.3%
Q	111.77	159.67	-42.9%	42.9%
R	148.37	198.27	-33.6%	33.6%
S	216.29	69.90	67.7%	67.7%
			MMRE	66.7%
			PRED(20)	30.0%
			PRED(30)	30.0%
			PRED(50)	60.0%

Table 3.20 shows the contractor calibrated CER data for Contractor 1, with MMRE and PRED(X) results.

Table 3.20. CER Calibrated by Contractor 1 Evaluation

Project	Actual Effort (P-M)	Estimated Effort (P-M)	Relative Error	Magnitude Relative Error
A	282.50	256.36	9.3%	9.3%
B	156.73	17.68	88.7%	88.7%
C	149.75	116.82	22.0%	22.0%
D	52.55	125.35	-138.5%	138.5%
E	18.32	21.30	-16.3%	16.3%
F	2.71	3.06	-12.7%	12.7%
G	5.69	32.15	-465.0%	465.0%
H	260.74	131.61	49.5%	49.5%
I	3.33	4.69	-40.8%	40.8%
			MMRE	93.6%
			PRED(20)	33.3%
			PRED(30)	44.4%
			PRED(50)	66.7%

Table 3.21 shows the contractor calibrated CER data for Contractor 2, with MMRE and PRED(X) results.

Table 3.21. CER Calibrated by Contractor 2 Evaluation

Project	Actual Effort (P-M)	Estimated Effort (P-M)	Relative Error	Magnitude Relative Error
J	1323.41	368.61	72.1%	72.1%
K	23.95	106.30	-343.8%	343.8%
L	336.68	354.01	-5.1%	5.1%
M	254.67	293.47	-15.2%	15.2%
N	61.31	41.99	31.5%	31.5%
O	69.67	66.13	5.1%	5.1%
P	101.51	162.10	-59.7%	59.7%
Q	111.77	162.06	-45.0%	45.0%
R	148.37	201.52	-35.8%	35.8%
S	216.29	70.59	67.4%	67.4%
			MMRE	68.1%
			PRED(20)	30.0%
			PRED(30)	30.0%
			PRED(50)	60.0%

Figure 3.11 shows the Relative Error of both the original base COCOMO II model, and the contractor calibrated COCOMO II cost model graphically to show the improvements made from calibration. We can see graphically that calibration truly improved the results from Contractor 1, while the scale of improvement in Contractor 2 is less obvious.

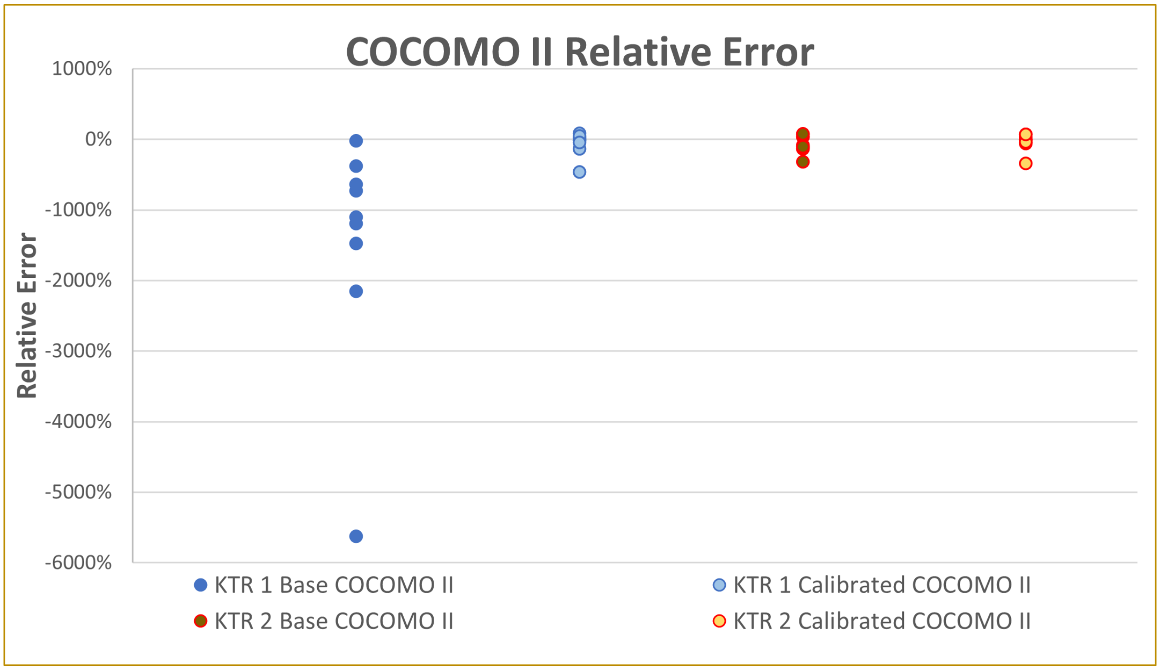

Figure 3.11. Relative Error of Base and Calibrated COCOMO II Models P-M

Figure 3.12 shows the Relative Error of both the original base CER models, and the contractor calibrated CER cost model graphically to show the improvements made from calibration. As with the COCOMO II calibration graph, the Contractor 1 calibration improvement is far more apparent, although the Contractor 2 improvement is also evident in this graph.

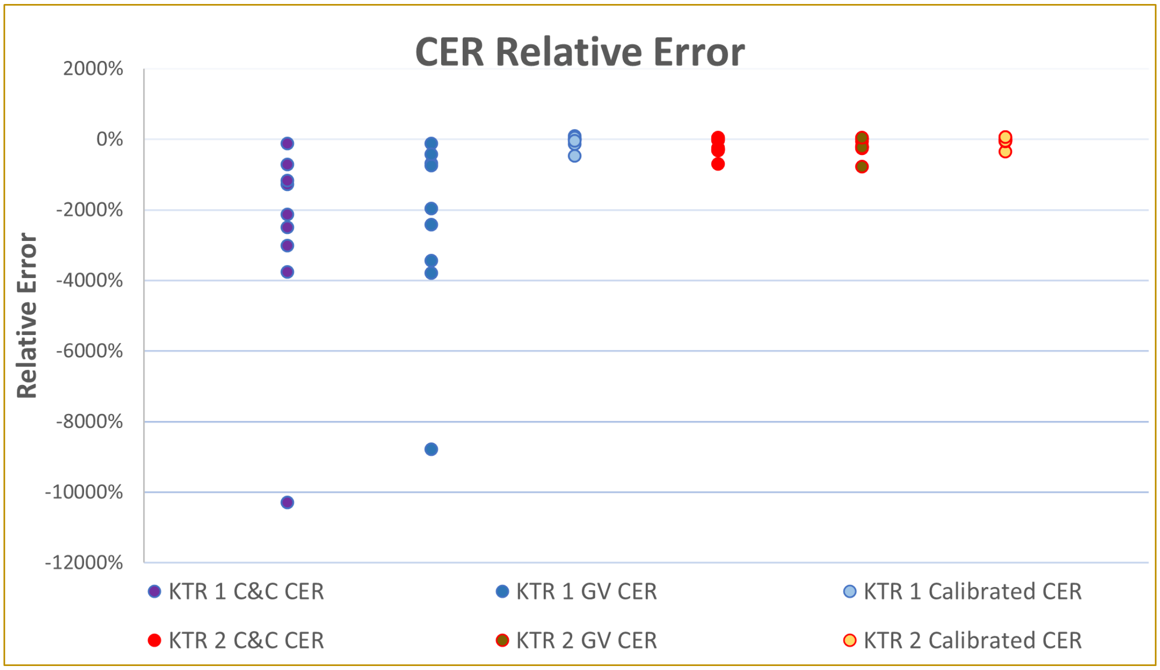

Figure 3.12. Relative Error of Base and Calibrated CER Models P-M

Figure 3.13 shows the Relative Error of each contractor calibrated cost model graphically to show the differences between the different contractors and methods. As with the uncalibrated data and the combined calibrated data, the data with low actual P-M are shown to have the most relative error.

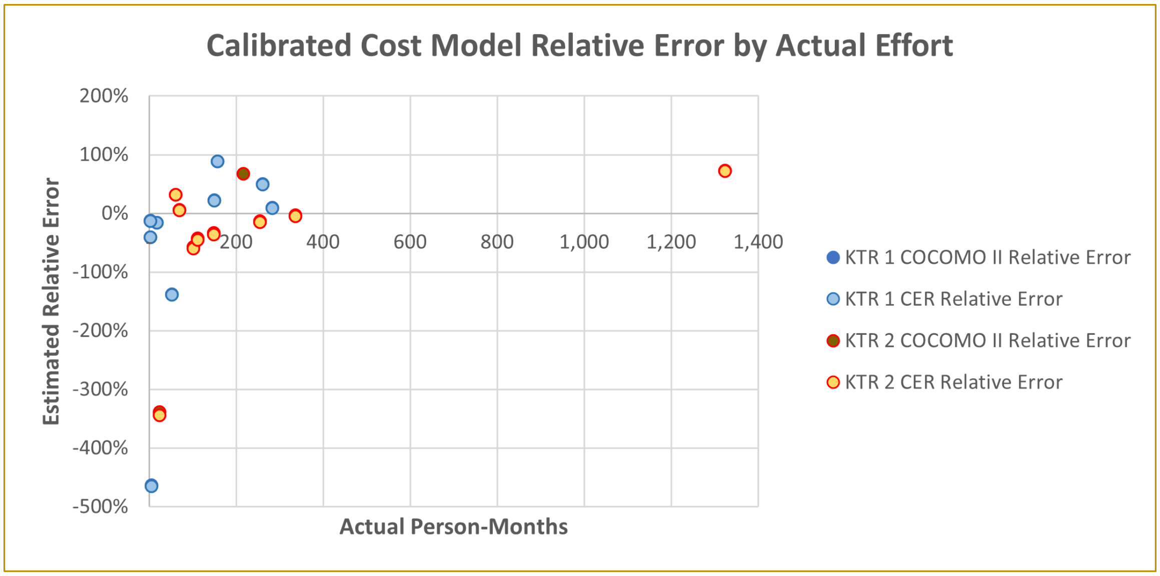

Figure 3.13. Scatter Plot of Cost Model Relative Errors Compared to Actual P-M

The data shows that both MMRE and PRED(30) values improved with contractor specific calibration for both COCOMO II and the CER results. In this case the contractor calibrated COCOMO II cost model had slightly better MMRE in each case, and had equivalent prediction accuracy for both contractors. The difference between the contractor calibrated COCOMO II results and the contractor calibrated CER results was not significant and could easily be due to rounding error in the equations.

3.8 Data Results

The team examined MMRE and prediction accuracies for 20%, 30%, and 50% magnitudes for all three data sets.

The resulting data shows that the best cost model for MMRE for the combined data set was the specialized calibrated COCOMO II, with an MMRE result of 76.2%. The results also show that the best cost model for MMRE for Contractor 1 was the specialized contractor calibrated COCOMO II, with an MMRE result of 93.3%. For Contractor 2, the best MMRE result was again the specialized contractor calibrated COCOMO II, with an MMRE of 66.7%. Out of the three MMRE measures, the best fit was for Contractor 2.

For prediction accuracy, the results show that for the combined data set, the specialized calibrated COCOMO II model had the best results, with the best prediction accuracies of PRED(20) = 15.8%, PRED(30) = 21.1%, and PRED(50) = 52.6%. For Contractor 1, the prediction accuracies were actually the same between the specialized contractor calibrated COCOMO II model the specialized environment calibrated CER, with prediction accuracies of PRED(20) = 33.3%, PRED(30) = 44.4%, and PRED(50) = 66.7%. For Contractor 2, the prediction accuracies were again a tie between the specialized contractor environment calibrated COCOMO II and the specialized environment calibrated CER, with prediction accuracies of PRED(20) = 30.0%, PRED(30) = 30.0%, and PRED(50) = 60.0%. For prediction accuracies, calibrating to the contractor provided the best results.

Combining both of these measurements, the best cost model tested for the combined data set was the calibrated COCOMO II model, as it has best MMRE and prediction accuracies across the board. The best cost model tested for Contractor 1 was the specialized contractor calibrated COCOMO II model, due to the better MMRE value and the prediction accuracies being equal. For Contractor 2, the best cost model was the specialized contractor calibrated COCOMO II model because it also provided the best MMRE result and had identical prediction accuracies.

Table 3.22 provides a summary of the different cost model MMRE measures and prediction accuracies for the combined data set. Figures 3.14 and 3.15 depict the measures for the combined data set. The results for Contractor 1 are summarized in Table 3.23. Figures 3.16 and 3.17 depict the measures for Contractor 1. Finally, the results for Contractor 2 are summarized in Table 3.24. Figures 3.18 and 3.19 depict the measures for Contractor 2.

Table 3.22. Cost Model Comparison For Combined Data

	COCOMO II	COCOMO II Cal	C&C CER	GV CER	CER Cal
MMRE	757.7%	76.2%	1429.7%	1280.2%	220.6%
PRED(20)	5.3%	15.8%	5.3%	5.3%	5.3%
PRED(30)	5.3%	21.1%	10.5%	10.5%	15.8%
PRED(50)	15.8%	52.6%	15.8%	15.8%	36.8%

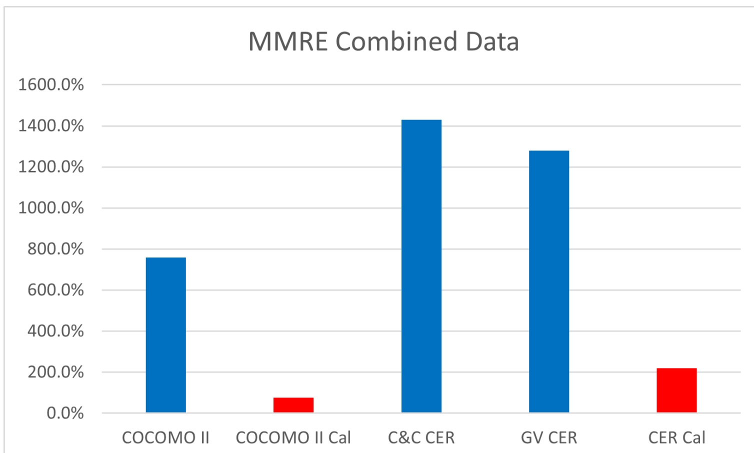

Figure 3.14. MMRE of All Tested Combined Cost Models

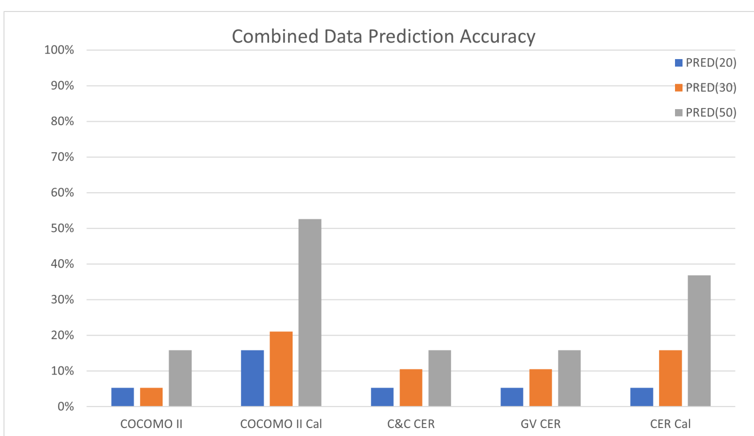

Figure 3.15. PRED(X) of All Tested Combined Cost Models

Table 3.23. Cost Model Comparison for Contractor 1 Data

	COCOMO II	COCOMO II Cal	C&C CER	GV CER	CER Cal
MMRE	1478.6%	93.3%	2770.9%	2480.9%	93.6%
PRED(20)	11.1%	33.3%	0.0%	0.0%	33.3%
PRED(30)	11.1%	44.4%	0.0%	0.0%	44.4%
PRED(50)	11.1%	66.7%	0.0%	0.0%	66.7%

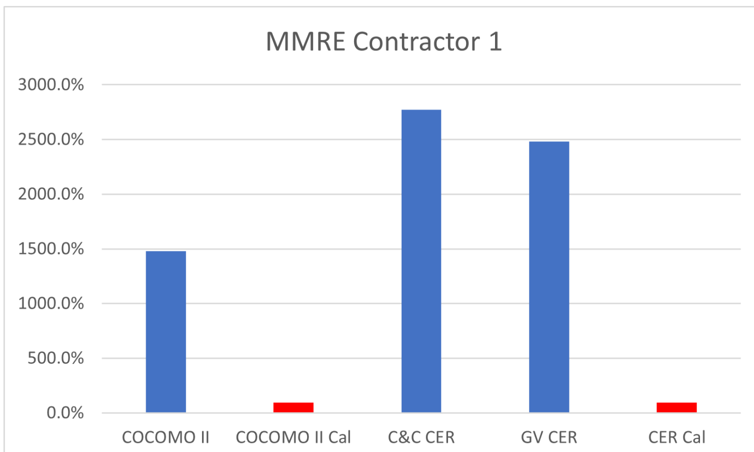

Figure 3.16. MMRE of All Tested Cost Models for Contractor 1

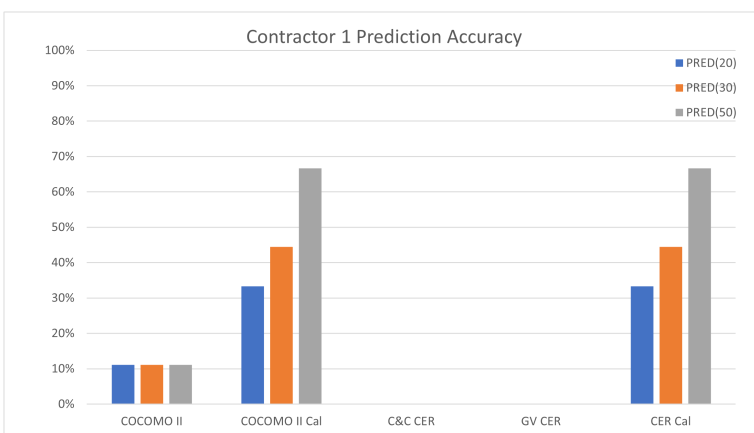

Figure 3.17. PRED(X) of All Tested Cost Models for Contractor 1

Table 3.24. Cost Model Comparison for Contractor 2 Data

	COCOMO II	COCOMO II Cal	C&C CER	GV CER	CER Cal
MMRE	109.0%	66.7%	222.7%	199.5%	68.1%
PRED(20)	0.0%	30.0%	10.0%	10.0%	30.0%
PRED(30)	0.0%	30.0%	20.0%	20.0%	30.0%
PRED(50)	20.0%	60.0%	30.0%	30.0%	60.0%

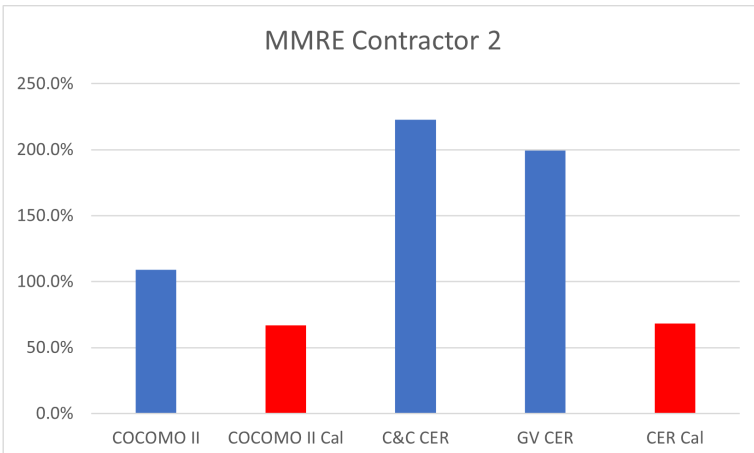

Figure 3.18. MMRE of All Tested Cost Models for Contractor 2

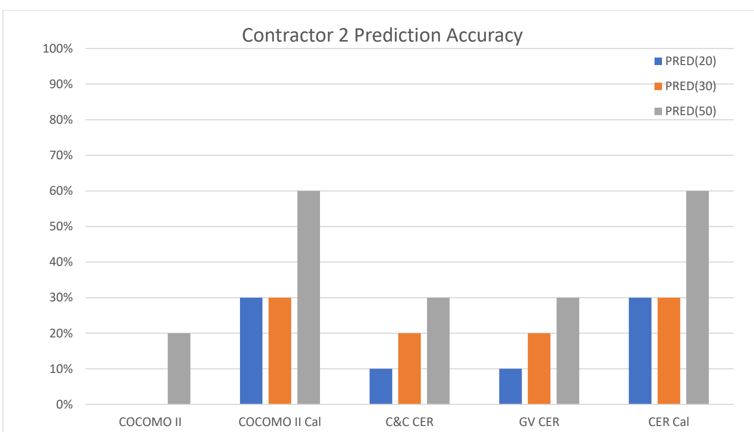

Figure 3.19. PRED(X) of All Tested Cost Models for Contractor 2

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Conclusions and Recommendations

4.1 Conclusions

Software cost estimating models allow for reproducible, standardized methods by which to plan a software project and validate an offeror's proposal. However, no model can predict efforts with complete accuracy. The team uncovered strengths and weaknesses within each model as well as recommended practices for real-world application.

Calibration of each model or method is vital to accurate estimates. After each model was calibrated, the MMRE measures were generally orders of magnitude better than the base model. The prediction accuracies also greatly improved across the board after calibration of each model, with some accuracies increasing from 0.0%. A summary of these improved measures can be found in Section 3.7. Further, calibrations can be tailored to a unique contractor, so there is potential to develop a standard calibration for each vendor. Since the data appeared to be two distinct sets of data by contractor, it is highly recommended to calibrate the data by contractor versus by the data as a whole; the contractor calibrated results were greatly improved from the combined data set measures, particularly for the prediction accuracies.

The specialized calibrated COCOMO II model proved to be the best fit model overall, with MMRE values of 76.2% (combined data set), 93.3% (Contractor 1), and 66.7% (Contractor 2). The calibrated COCOMO II model proved to be the best fit again with the prediction accuracies, with PRED(20) = 15.8%, PRED(30) = 21.1%, and PRED(50) = 52.6% for the combined data set. For Contractor 1, the calibrated COCOMO II model and the calibrated CER had identical prediction accuracies of PRED(20) = 33.3%, PRED(30) = 44.4%, and PRED(50) = 66.7%. For Contractor 2, there was again a tie between the calibrated COCOMO II model and the calibrated CER, with PRED(20) = 30.0%, PRED(30) = 30.0%, and PRED(50) = 60.0%. The team believes that if the COCOMO scale and cost factors were adjusted away from nominal, there would be more of a prediction accuracy difference noted between calibrated COCOMO II and the calibrated CER.

4.2 Recommendations

In conducting the background research, acquisition community survey, and data analysis for this capstone project, the team generated a number of suggestions for improvement that could increase the accuracy of software acquisition project cost estimates. These suggestions pertain to both the methods and models examined in this research and organizational changes for better business practices. The below outline these suggested improvements.

4.2.1 Utilizing Checklists

The team highly recommends the use of a standardized checklist (or checklists) when initializing an effort to use automated cost estimating models for software projects, either to estimate the project or to validate an offeror's estimate.

Magne Jørgensen produced "A Preliminary Checklist for Software Cost Management" [21] and Robert Park produced "A Manager's Checklist for Validating Software Cost and Schedule Estimates" [22] and "Checklists and Criteria for Evaluating the Cost and Schedule Estimating Capabilities of Software Organizations" [23]. All are excellent sources for standard checklists that have already been developed.

Throughout the conduct of this capstone, the team developed and refined a checklist that can guide the creation and validation of a software cost estimation model. The detail of this checklist can be found in detail in section 3.2.

The resulting calibrated model would be used to estimate future software projects. The project team would need project requirements and subject matter expertise to estimate the size (in ESLOC) of a future effort and use the calibrated model to produce an estimate of the project effort.

4.2.2 Model Tailoring and Calibration

The team observed marked improvement in the performance of each estimating model after it was calibrated to the unique environment. Further, having enough vendor data to be able to adjust the effort multipliers and scale factors within COCOMO II would have allowed for further refinement and better accuracy from this model. Also, COCOMO II calibration using only the multiplicative constant, and not the baseline exponents would have gotten different results if they were attempted (the calibration for this project adjusted both variables). Calibration of any model to each system, software project, or contractor environment would assist in realizing the goal of best fitting an effort.

4.2.3 Scale and Cost Drivers Expertise

In conducting software cost estimating efforts on the DoD side of a project, it is logical to utilize similar or the same estimating models as the offeror. Part of this includes understanding and maximizing the use of the standard scale and cost drivers within the model. There should be expert understanding within the DoD team of how these scale and cost drivers affect the estimate, in staffing requirements, level of effort, cost, and schedule. This level of understanding would allow the DoD to be extremely effective in cross-checking a vendor proposal for fairness and accuracy. Moreover, these scale and cost drivers could likely be standard per vendor or program. Meaning, once the initial expert review is conducted and these drivers established, there is potential for them to be reused with minimal editing for each subsequent effort within a program office, on the system, or with that vendor.

4.2.4 Standardization of SRDR Data

After reviewing the data submitted by both contractors, there are obvious differences in the data sets above and beyond what could be considered variations due to contractor expertise or capabilities and their interactions with the scope of the projects. Much of this study shows clear differences between each other as well as the assumptions and industry benchmarks used to create the base COCOMO II model and other CERs.

These reporting discrepancies are consistent throughout the data of Contractor 1. The team believes the data was reported according to the contractor's understanding of how to report data on SRDR forms. Further standardization and clearer definitions of data required on SRDR data could assist the contractors in reporting the data correctly.

All the SRDRs reported non-SLOC producing work hours (administrative hours) on the same WBS as SLOC producing hours, forcing the team to normalize this data. If all contractors had reported data properly, this step could have been skipped and the estimate may have been more accurate.

Many of these projects are multi-year projects with changing periods of performance; this may be the cause of at least some of the anomalies found in the data. It is possible the contractor is not receiving clear guidance for such situations that are not found in the SRDR Implementation Guide, DOD5000.04-M [25], and Data Item Description DI-MGMT-82035A [26].

4.2.5 Requirements Refinement

DoD systems acquisition efforts establish and track requirements at the systems level. This can make it difficult to isolate, analyze, and track software requirements solely. To conduct this research capstone, the team needed to conduct extensive survey research to find individual software development and maintenance projects within the Army portfolio. As all systems become more sophisticated and reliant upon software, particularly in the upgrade of capabilities, it would be very beneficial to the DoD to begin refining requirements for software rigorous systems down to the software level. Even considering the software portion of a system as its own separate system with separate requirements that can be traced up to the system level could prove cost effective and beneficial in the out-years of supporting the system.

4.2.6 Regulatory Requirements

Some of the efforts explored in the data analysis were for systems in sustainment. Those efforts are supported by Sustainment System Technical Service (SSTS) contracts. Until recently, CSDRs were not required on SSTS contracts, nor were they required on programs below Acquisition Category (ACAT) I and IA. These reports are the primary means that the DoD uses to capture actual cost data from completed efforts. If they were required for all programs, the data collected could inform future efforts and aid in future calibration of the models. Therefore the team recommends collecting this data on all future efforts. However, the team recognizes that this would be a far-reaching process shift, and it may not be cost effective for all programs.

Luckily there have been some recent changes in this area. A memo from the Director of Cost Assessment and Program Evaluation for the United States Department of Defense has expanded the CSDR requirements to cover ACAT II contracts as of January 2019. Since this is a recent change, any contract signed before this date would not have these requirements so it is still difficult to find historical data due to this. In addition, the CSDR requirements became codified into law as of 2017, as 10 USC 3227 was updated to require all DoD contracts to include cost tracking. In addition, SSTS contracts are now being required to contain CSDR plans when they meet a threshold of $50M as the law does not specify what types of contracts are required to track costs. These are new changes and are only now starting to trickle into new contracts.

4.2.7 Analytical Centers of Excellence

Many of these suggested improvements focus on an increase in institutional knowledge of software cost estimating within the DoD. The acquisition community is moving into a technologically savvy and software heavy environment. Logic dictates that the knowledge and skills required to verify and validate the efforts on these programs must also evolve. However, educating the whole DoD acquisition community would be costly and time consuming. Targeting this education to a pool of expert analysts that reside within each PEO would accomplish this goal much quicker and create knowledgeable centers of excellence that support similar systems. As programs move through their acquisition cycles, they can pull from this team of experts as needed, and all program offices within a PEO could share this resource. These analysts would not be experts in each program, but experts in analyzing software efforts, and could support programs with minimal background knowledge on the system being acquired.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] M. Jørgensen and M. Shepperd, “A systematic review of software development cost estimation studies,” *IEEE Transactions on Software Engineering*, vol. 33, no. 1, pp. 33–53, 2006.

[2] B. W. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. J. Reifer, and B. Steece, *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ: Prentice Hall Press, 2009.

[3] R. Madachy, “COCOMO II - Constructive Cost Model,” Sep. 01, 2022 [Online]. Available: http://softwarecost.org/tools/COCOMO/

[4] B. Clark and R. Madachy, *Software Cost Estimation Metrics Manual for Defense Systems*. Haymarket, VA: Software Metrics Inc., 2015.

[5] M. Jørgensen, B. Boehm, and S. Rifkin, “Software development effort estimation: Formal models or expert judgment?” *IEEE Software*, vol. 26, no. 2, pp. 14–19, 2009.

[6] M. Jørgensen, “Practical guidelines for expert-judgment-based software effort estimation,” *IEEE Software*, vol. 22, no. 3, pp. 57–63, 2005.

[7] *Department of the Army Cost Analysis Manual*, Department of the Army, Washington, DC, USA, 2002.

[8] M. Cohn, “Estimating with use case points,” *Methods & Tools*, vol. 13, no. 3, pp. 3–13, 2005.

[9] J. Rashid, M. W. Nisar, T. Mahmood, A. Rehman, and S. Y. Arafat, “A study of software development cost estimation techniques and models,” *Mehran University Research Journal of Engineering and Technology*, vol. 39, no. 2, pp. 413–431, 2020.

[10] M. Jørgensen, “What we do and don’t know about software development effort estimation,” *IEEE Software*, vol. 31, no. 2, pp. 37–40, 2014.

[11] *DoD Cost Estimating Guide., Version 1.0.*, Office of the Secretary of Defense, Cost Assessment and Program Evaluation, Washington, DC, USA, 2020.

[12] Y. Yang, D. Falessi, T. Menzies, and J. Hihn, “Actionable analytics for software engineering,” *IEEE Software*, vol. 35, no. 1, pp. 51–53, 2017.

[13] J. Hihn and G. Tregre, “Assuring software cost estimates: Is it an oxymoron?” in *2013 46th Hawaii International Conference on System Sciences*. IEEE, 2013, pp. 4921–4929.

[14] *Software Development Cost Estimating Handbook*, Air Force Cost Analysis Agency, Arlington, VA, USA, 2009.

[15] K. Lum, M. Bramble, J. Hihn, J. Hackney, M. Khorrami, and E. Monson, “Handbook for software cost estimation,” *Jet Propulsion Laboratory, Pasadena, CA, USA*, 2003.

[16] *Federal Acquisition Regulation*, General Services Administration, Department of Defense, and National Aeronautics and Space Administration, Washington, DC, USA, 2019.

[17] *Defense Federal Acquisition Regulation Supplement*, Department of Defense, Washington, DC, USA, 2020.

[18] “Calico for calibration,” Softstar Systems, 2020 [Online]. Available: http://www.softstarsystems.com/calico.htm

[19] D. Ligett, “Techniques for calibrating cocomo estimating models,” http://www.softstarsystems.com/calibrat.ppt, 2001.

[20] *COCOMO II - Model Definition Manual*, 2nd ed., Center for Software Engineering, USC, Los Angeles, CA, USA, 1995 - 2000.

[21] M. Jorgensen and K. Molokken, “A preliminary checklist for software cost management,” in *Third International Conference on Quality Software, 2003. Proceedings.* IEEE, 2003, pp. 134–140.

[22] R. E. Park, “A manager’s checklist for validating software cost and schedule estimates.” CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, Tech. Rep., 1995.

[23] R. E. Park, “Checklists and criteria for evaluating the cost and schedule estimating capabilities of software organizations.” CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, Tech. Rep., 1995.

[24] R. Madachy, private communication, Jul. 2022.

[25] *SRDR Implementation Guide, DOD 5000.04-M*, Department of Defense, Washington, DC, USA, 2019.

[26] *Data Item Description, DI-MGMT-82035A*, Department of Defense, Washington, DC, USA, 2017.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California