9-1997

Calibration and Validation of the Sage Software Cost/Schedule Estimating System to United States Air Force Databases

David B. Marzo

AFIT/GCA/LAS/97S-6

CALIBRATION AND VALIDATION OF THE SAGE SOFTWARE COST/SCHEDULE ESTIMATING SYSTEM TO UNITED STATES AIR FORCE DATABASES

THESIS

David B. Marzo
Captain, USAF

AFIT/GCA/LAS/97S-6

Approved for public release; distribution unlimited

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, the U.S. Government, or the model developer.

AFIT/GCA/LAS/97S-6

CALIBRATION AND VALIDATION OF THE SAGE SOFTWARE COST/SCHEDULE ESTIMATING SYSTEM TO UNITED STATES AIR FORCE DATABASES

THESIS

Presented to the Faculty of the Graduate School of Logistics
and Acquisition Management of the Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Cost Analysis

David B. Marzo, B.S.
Captain, USAF

September 1997

Acknowledgments

The past 15 months have presented some of the most challenging moments of my military career, as well as some of the most rewarding. I learned early in this master's program that the ability to succeed is very much dependent upon having a strong team around you. I am so fortunate to have benefited from three wonderful teams: my classmates, the faculty, and my family.

My classmates in the Graduate Cost Analysis (GCA) program have absolutely been the best group of people I have worked with in the military. They made this program great for me. My sincerest thanks, respect, and admiration to Wayne Bernheisel, John Cole, Chris Dalton, Bill Forster, Judd Fussell, Mark Glenn, Seon-mook Lee, Tom Shrum, Mark Sweitzer, and Tom Van Egeren. I add a special thanks to Wayne Bernheisel and Tom Shrum, who authored theses of a similar nature to mine and who helped me considerably through this process. I hope that I have the honor to work with each of you gentlemen again.

I also extend my appreciation to a superb faculty, specifically my thesis advisor, Professor Daniel Ferens and reader, Dr. David Christensen. Your insights, comments, and ability to give me responsive, constructive criticism made this effort rewarding. I have appreciated your continued support and encouragement. Thanks also to Lt Col Stephen Giuliano, the GCA Program Director. You are a great role model, not only as a cost analyst, but also as an Air Force officer.

Thanks so much to my family: Jill, my wife, Ryan, my son, and Mackenzie, my daughter. I have not always been the best husband or father over the past few months, but you always were there to push me along and give me plenty of love and support. Thanks also to my parents who instilled in me the work ethic and heart to get through all of these challenges.

A final thanks to Dr. Randy Jensen, who graciously devoted hours of his time to help me perform this effort and provided invaluable insight into the nature of software development and cost estimation.

David B. Marzo

Table of Contents

Page

Page

List of Figures

List of Tables

Table Page

List of Equations

Abstract

This research entailed calibration and validation of the SAGE Software Cost/Schedule Estimating System, Version 1.7 as a means to improve estimating accuracy for DoD software-intensive systems, and thereby introduce stability into software system development. SAGE calibration consisted of using historical data from completed projects at the Space and Missile Systems Center (SMC) and the Electronic Systems Center (ESC) to derive average performance factors (i.e., calibration factors) for pre-defined categories of projects. A project was categorized for calibration by either its primary application or by the contractor that developed it. The intent was to determine the more appropriate categorization for calibration. SAGE validation consisted of using the derived calibration factors to predict completed efforts, not used in deriving the factors. Statistical resampling employing Monte Carlo simulation was used to calibrate and validate the model on each possible combination of a category's projects. Three statistical measures were employed to measure model performance in default and calibrated estimating modes. SAGE generally did not meet pre-established criteria for estimating accuracy, although the model demonstrated some improvement with calibration. Calibration of projects categorized by contractor resulted in better calibrated model performance than calibration of projects categorized by application. This categorization is suggested for future consideration.

CALIBRATION AND VALIDATION OF THE SAGE SOFTWARE COST/SCHEDULE ESTIMATING SYSTEM TO UNITED STATES AIR FORCE DATABASES

I. Introduction

Overview

Computer software has achieved a pervasive influence in many, if not all, facets of Department of Defense (DoD) operations. One Air Force general officer recently commented that, "The only thing you can do with an F-22 that does not require software is to take a picture of it" (Department of the Air Force, 1996: 2-6). The rate at which software has gained this influence has been phenomenal. To put it in perspective, consider the case of the F-4 Phantom, a Vietnam War vintage jet that used no software in its weapon systems (Department of the Air Force, 1996: 2-5). A portion of this DoD software revolution was inevitable considering the rapid advances in computer processing power in recent decades; however, it is also fair to characterize the DoD as a proactive participant in seeking opportunities to tap software's power. The federal budget provides evidence to this as the DoD committed almost 17% of its FY95 $252 billion budget to software, an increase of 31% in just three years (Department of the Air Force, 1996: 2-15). Managing this software explosion, like managing any rapid-growth enterprise, poses a formidable challenge to DoD managers.

Mosemann suggests that the biggest concern for the DoD with respect to software is a collective inability to predict its costs and schedule (Department of the Air Force,

1996: i). The literature on software cost estimates is replete with stories of cost estimates that are orders of magnitude different from actual costs. Humphrey punctuates this with his claim that, "Regardless of the methods used, software estimates will be inaccurate" (Humphrey, 1990: 92). In quantitative terms, the current state of the DoD is that software cost estimates are accurate within 25% of actuals only about half of the time (Ferens, 1996: 31).

The consequence of unpredictable software effort is the instability of the acquisition program for which the software is being developed. In the DoD, the cost estimate becomes the basis for the budget. When actual effort significantly exceeds that estimated, problems can arise because resources are inadequate to perform. One text describes a vicious cycle in which corners are cut to keep costs down and this ultimately leads to an attempt by the developing organization to deliver less for the same price (Department of the Air Force, 1996: 1-21 - 1-22). Perhaps this cycle lends insight into Gibbs' claim that almost three quarters of all large software-intensive systems either do not function as intended or are not used at all (Gibbs, 1994: 87). Because of this instability, considerable effort has been dedicated to improving software cost estimation. This research examines efforts to improve software estimating accuracy within the DoD and specifically examines calibration of the SAGE Cost/Schedule Estimating System for use in the United States Air Force.

General Issue

The DoD primarily relies on parametric models to estimate software costs. Some of the more common models currently in use include REVIC, SEER-SEM, CHECKPOINT, SLIM and PRICE-S. The DoD prefers parametric models because these models are relatively easy to use and require fewer people to perform an estimate than

other common estimating methods; however, parametric models have historically been inaccurate (Ferens, 1996: 29-31).

Parametric models generate estimates on the assumption that software cost or effort is a mathematical function of any number of inputs into the development process. Typical inputs that many models consider are software size (e.g., lines of programming code) and the capabilities of the people and organization who are performing the effort. The estimator enters available project information into the model and the model calculates an estimate based on these inputs. The estimator can then update the estimate when more project information is available. This process is less demanding than one that requires the estimator to formulate cost estimating relationships and build an estimate entirely from scratch. Despite this ease, estimating accuracy remains an elusive goal.

Many experts suggest calibration as a means to improve the accuracy of parametric model equations. Calibration involves adjusting the model's effort equations for a particular environment. The default model equations are derived in part or in whole from a historical database of programs. These equations are therefore dependent upon and reflect the database from which they were derived. For example, a model formulated from a history of military programs in theory generates estimates that are reflective of the military environment. Yet, even within the military, there is a wide variety of demands placed on software systems. For example, the requirements placed on an embedded software system for space are much more stringent than those placed on a stand-alone management information system. The difference in effort required to develop two such disparate systems may be considerable. Although a model may be marketed for DoD cost estimating purposes and may capture macro-level differences in system complexity, the model is typically not fine tuned for specific DoD environments. Calibration of the model's equations is therefore suggested as the means to adjust the model to improve its estimating accuracy for a particular environment.

Specific Issue

This thesis details the calibration of the SAGE Cost/Schedule Estimating System. SAGE is a relatively new estimating model that is developed and marketed by Dr. Randall Jensen through his company, Software Engineering, Incorporated. This thesis continues a body of research started in 1995 at the Air Force Institute of Technology (AFIT) to calibrate common cost estimating models to improve model estimating accuracy for the Air Force. This AFIT research stream includes seven previous and three concurrent calibration efforts that will be summarized in Chapter II, Literature Review. For this effort, SAGE is calibrated to historical data provided from both the Electronic Systems Center (ESC) and the Space and Missile Systems Center (SMC) of the Air Force Material Command.

The decision to calibrate to two databases is a departure from recent AFIT research that used data only from SMC. The SMC database is used in response to their requirement to achieve a calibrated cost estimating ability for SAGE. In fact, separate calibration factors are derived for a variety of program categories at SMC such as space, avionics, and command and control. The ESC database is employed because it includes unique information that enables calibration to be performed on a contractor-by-contractor basis. This is important because Dr. Jensen's belief, as reflected in SAGE, is that the variation in development effort, for a given program size and complexity, is driven largely by management factors unique to a company. By deriving calibration factors for a specific company, insight is gained into how efficiently the company develops software. This is in contrast to the method employed for the SMC data in which performance is calibrated at an aggregated industry level of performance. By calibrating the model to both databases, an assessment can be made as to the more effective method.

Research Contribution

This effort represents the first time that the SAGE Cost/Schedule Estimating System has been calibrated to either the ESC or SMC database. Apart from model choice, this research is unique from previous AFIT efforts in that calibration is accomplished against data stratified primarily by contractor.[1] This research also introduces the use of simulation to perform calibration and validation such that all subsets of project data are considered for calibration and validation, versus the standing practice of considering just one subset of the data. This results in much more robust findings.

Research Objectives

The research objectives are as follows:

1. To critically examine previous efforts to calibrate software cost models for the DoD and, where possible, to improve upon the general methods used.
2. To establish a specific methodology for SAGE model calibration.
3. To measure the improvement, if any, from calibrated SAGE estimates as compared to uncalibrated SAGE estimates.
4. To determine whether the model meets the criteria of an effective estimating model.
5. To examine whether calibrating the model to a specific contractor's capabilities is more appropriate than calibrating the model to a general class of programs.
6. To propose future research efforts in this area.

[1] Concurrently, Shrum, 1997 is performing a calibration of the CHECKPOINT model that considers calibration primarily by contractor.

Scope of Research

This research is limited to calibration of the SAGE Cost/Schedule Estimating Model, Version 1.7 to improve its ability to estimate for development effort only. Calibration of the model for schedule estimation is not explicitly performed; however, improvements in schedule estimating may be implicitly coupled with improved estimates for effort. No data points other than those provided from the ESC and SMC databases are used for either calibration or validation. As such, use of the calibration results for programs external to these specific domains is cautioned.

Thesis Overview

This chapter provides the motivation for the need to calibrate software cost estimating models to improve their accuracy. Model estimates have historically been inaccurate, and inaccurate estimates translate into unstable programs. The consequence of this instability is a vicious cycle of overruns and reduced capability. By improving estimating accuracy through calibration of a model's equations, estimators can introduce stability to the programs and thereby increase the chance of program success. This success is vital to the DoD, in view of the heavy emphasis that the DoD places on software-intensive systems.

Chapter II, Literature Review, provides a summary of related calibration efforts and offers an analysis of what the literature suggests as ways to improve estimation and the calibration process. In addition, a thorough analysis of the underlying principles and mathematical equations of the SAGE Cost/Schedule Estimating model is offered.

Chapter III, Methodology, details the steps in calibrating and validating SAGE. Focus is given to describing the databases that are used for calibration, identifying the SAGE parameters that will be calibrated, and explaining the measures used to assess the model's accuracy for both a calibration and a validation data set.

Chapter IV, Findings, describes the results of the calibration and validation efforts in terms of the statistical measures proposed in Chapter III.

Chapter V, Conclusions and Recommendations, reviews the findings of Chapter IV and analyzes the degree to which the stated research objectives have been achieved. Future directions for related research are offered.

II. Literature Review

Overview

This chapter examines the nature of software cost estimation in order to describe the context in which the SAGE model is used. Specifically, the chapter reviews the estimating methodologies that are generally available and which of those are typically employed by Department of Defense (DoD) cost analysts. A review of efforts taken to improve these estimating methods is also offered. The chapter then focuses on the background, philosophy, and behavior of the SAGE model. The intent is to establish a foundation for understanding the model, and on which a calibration and validation methodology will be established.

Software Cost Estimating Methodologies

Several cost estimating methodologies are available to the estimator. The SAGE model, for example, employs a parametric or algorithmic method since it forecasts development effort as a mathematical function of inputs such as the size of the program and the capabilities of the developing team. Boehm identifies the algorithmic method as one of seven available estimating techniques (Boehm, 1981: 329-330). These techniques are summarized in Table 1 on the following page.

The issue for the estimator then becomes to decide which of these methodologies to employ. Boehm says the Parkinson and price-to-win methods do not produce sound cost estimates and are unacceptable (Boehm, 1981: 334). Of the other methods, Boehm says that none of the alternatives is better than the others from all aspects and the strengths and weaknesses of these techniques are complementary (Boehm, 1981: 334). The choice of methods thus hinges on the situational and environmental factors in which the estimator is working.

Table 1. Cost Estimating Methodologies

Method	Description
1. Algorithmic Models	Provide one or more algorithms that produce a software cost estimate as a function of a number of variables which are considered to be the major cost drivers
2. Expert Judgment	Involves consulting one or more experts, perhaps with aid of an expert-consensus mechanism such as the Delphi technique.
3. Analogy	Involves reasoning by analogy with one or more completed projects to relate their actual costs to an estimate of the cost of a similar new project.
4. Parkinson	The principle that work expands to fill the available volume is invoked to equate the cost estimate to the available resources.
5. Price to Win	The estimate is equated to the price believed necessary to win the job.
6. Top-Down	An overall cost estimate for the project is derived from global properties of the software product. The total cost is then split up among the various components.
7. Bottom-up	Each component of the software job is separately estimated, and the results aggregated to produce an estimate for the overall job.

(Boehm, 1981: 329-330)

Several factors must be taken into account. One factor is the point of the program at which the estimate is accomplished. For example, Wellman points out that the bottoms-up approach cannot be done "until there is a well-defined design and the nature and size of the components are known" (Wellman, 1992: 31). A second factor is the required stability of the estimate. With respect to stability, Ferens critiques the algorithmic models because "they are often unstable, in that small changes in certain sensitive input parameters can result in substantial changes in cost or schedule" (Ferens, 1996: 29). Those who desire stable estimates may instead consider a bottoms-up approach (Boehm 1981: 342). A third factor is the type or nature of the program being estimated. Stutzke says that, "No single estimating method is suited for every type of project" (Stutzke, 1996: 20). For example, if software is being developed for a brand new application, then it would be inappropriate to use the analogy method since no comparable program history exists.

It is incumbent on the estimator to evaluate the particular situation and environment and then decide which method(s) to use. The estimator should also look at estimation as a dynamic process. Stutzke recommends a "regular review of progress, assumptions, and product requirements during a project to detect changes and violations of the assumptions underlying the estimate" (Stutzke, 1996: 20-21). For example, the estimator may make an initial estimate using a parametric model and then, as more information becomes available, use a bottoms-up methodology to generate a more stable, accurate estimate.

Considering these factors, the DoD's tendency to rely on algorithmic software cost estimating methods can be examined. Ferens indicates that this reliance is driven by several elements. He says DoD analysts typically do not have an ongoing, in-depth knowledge of the projects they are estimating, but they must frequently perform thorough estimates in a very short period of time. In addition, DoD analysts must usually accomplish their estimates early in the project (Ferens, 1996: 29). Each of these factors play into the strengths of algorithmic models because these models allow the estimator to generate relatively quick estimates on a rather limited amount of information. This is not to imply that software algorithmic models are the only estimation tools employed by DoD cost analysts; however, the DoD's heavy use of these models warrants insight into how they might be better employed for DoD estimating.

Algorithmic Estimating Models

General Description. There are approximately 50 commercial software estimating models available in the United States and another 25 marketed abroad (Jones, 1996: 19). The variety of estimating approaches represented by each of these models lends credence to Jones' claim that software cost estimating is a very difficult intellectual

problem. Despite this complexity, Jones claims these models generate estimates on three following fundamental equations (Jones, 1996: 20-21):

Staff = Size of deliverable/assignment scope (1)

Effort = Size of deliverable/productivity rate (2)

Schedule = Effort/staff (3)

From a macro perspective, these equations make logical, simple sense. For example, in Equation 1, the program staff is computed by dividing the size of deliverable by the amount or scope of work particular persons will add to the effort. Equation 3, the program schedule is derived by simply dividing total effort by the staff available to perform the effort. At a micro level, however, these models can diverge considerably. It is instructive to demonstrate this divergence by dissecting Equation 2 for effort. Through this dissection, additional insight is gained into the estimating philosophy of algorithmic estimating models.

Dissecting the Fundamental Effort Equation. Equation (2), Jones' fundamental equation for effort, can be dissected by examining each of its elements and reviewing the differences in how various algorithmic models consider these elements. A main distinction in models arises in the scope of the effort being computed. For example, the COCOMO model estimates the effort it takes to develop a program from preliminary design through component test, whereas the PRICE-S model's range is more expansive as it estimates the effort required from initial system requirements to system test (Ferens, 1997: 3-7, 4-2). A key element for the model user is to understand what the estimate is providing and to clearly explain this underlying basis to those who will use the estimate for decision-making.

There are also considerable differences in methods used to measure the size of deliverable. The two primary measures of size are source lines of code (SLOC) and function points. Most SLOC measurements capture the size of a program by counting all executable instructions and data declarations but exclude comments, blanks, and continuation lines (Department of the Air Force, 1996: 8-34). The strength of using SLOC is that it is a common measure in practice and in estimating models. Over time, relatively standardized definitions of SLOC have emerged. On the other hand, some criticize SLOC as a measurement because of the "lack of international standards that clearly defined what was meant by a 'line of code' in any common language...," (Jones, 1991: 8). The challenge of using SLOC is compounded by the difficulty of quantifying it early in the program based on initial requirements. The estimator is thereby challenged to update estimates as more information is available. "Only through this constant re-evaluation can the predictive model provide a cost that approximates actuals" (Department of the Air Force, 1996: 8-34).

On the other hand, a function point, as defined by Capers Jones, "is an abstract but workable surrogate for the goods that are produced by software projects" (Jones, 1991: 46). Specifically, function points are the weighted sums of the following five factors that relate to user requirements: inputs, outputs, logic files, inquiries, and interfaces (Jones, 1991: 46). Function points are primarily used to measure the size of management information systems and feature points, which are similar to function points, are used for system software (Jones, 1991: 9). The strengths of using either function points or feature points is that they are specification-based, language independent, and user-oriented. The drawback is that, like SLOC, they are difficult to estimate and that accounting for function points can be subjective and inconsistent between estimators (Department of Air Force, 1996: 8-33 - 8-36).

Finally, measuring the productivity rate is probably the most subjective element of the effort equation. In general, the models agree that productivity is a function of factors such as the development team's collective software development experience, education, work environment and use of modern practices and development tools. Differences arise, however, in which of these factors are relatively more important, and specifically how these factors are measured. Again, it is incumbent on the estimator to understand the basis of the model's operating parameters and learn how these parameters compute in the overall estimate.

In fact, a common theme in the above discussion is the vital role that the estimator plays in understanding and using the model. The estimator must be pro-active in understanding the model's basic equations and assumptions. The estimator's next step is to fine-tune the model as needed for the environment in which he or she is working. This leads to the following discussion on efforts taken to improve software cost estimating through a process of model calibration and validation.

Efforts to Improve Use of Software Algorithmic Cost Estimating Models

General Philosophy. Efforts to improve algorithmic software models have centered on a process of calibration and validation. As discussed in Chapter I, calibration is the adjustment of a model's basic estimating equations so that the model provides better estimates for a particular environment. Two general methods have been advanced to do this. Both involve the use of historical data from the environment in question. The first method is to compute an average or composite value for one of the independent variables in the model's estimating equation and then use that value as a constant in estimating future similar efforts. Typically, a factor that measures historical productivity is calibrated. The second method assumes that the model's estimating equations are

known and then a process can be undertaken to adjust the equation's coefficient or exponential weights (Van Genuchten and Koolen, 1991: 41).

Consider, as an example of the first method of calibration, the theoretical calibration of Equation 2, Jones' fundamental equation for effort.

Effort = Size of deliverable/productivity rate (2)

Calibration for this equation would occur by deriving a composite or average productivity rate that has occurred over time within a given development environment. In practice, this would be done by plugging in the known, historical values for program effort and size and then solving the equation for the productivity factor. For instance, a program that took fifty months of effort and contained 50 thousand SLOC would yield a computed productivity factor of 1,000. This process is repeated for each program in which historical information is known. The productivity factors for each of these historical programs are then averaged, which results in a composite calibration factor. This factor is then plugged in as a constant value to the model equation as the model is used to forecast other efforts within that environment.

The second method, that of adjusting the model's coefficient or exponential weights, must be applied more cautiously. In some cases, it is not applicable because the model's estimating equations may not even be known. If the model's equations are known, then a sufficient amount of data still must be available to generate a stable calibration. Van Genuchten and Koolen point out, "For example, if we wish to calibrate the COCOMO model with approximately 75 weights attributed to 15 variables, then a lot of data are necessary to adopt the weights in an accountable manner" (Van Genuchten and Koolen, 1991: 41). Boehm suggests that at least historical data from at least 10 projects should be available before simultaneously calibrating the coefficient and exponent of a COCOMO equation; if less than 10 are available, Boehm recommends calibration of the coefficient term only (Boehm, 1981: 529). Also, note that calibration

by adjusting the model's coefficient or exponential weights is essentially equivalent to creating a new estimating model.

Regardless of the method employed, the calibrated model should, in theory, yield better estimates than an uncalibrated model, at least for the set of programs used to perform the calibration. This is essentially because the calibration was geared to improving estimation for that specific data. The true test of calibration effectiveness is, therefore, validation, in which the calibration factor is used to estimate programs that were not included in the calibration data set. Statistical measures can then be used to compare the model's improvement in estimating accuracy for these programs when the model is run with the calibrated factor and when it is not. The effectiveness of calibration can then be assessed.

AFIT Calibration and Validation Efforts. A series of studies was begun in 1995 at AFIT under the guidance of Professor Daniel Ferens and Dr. David Christensen to calibrate and validate algorithmic models used by the Air Force, specifically at the Space and Missile Systems Center at Los Angeles Air Force Base. The models that have been calibrated include REVIC, SEER-SEM, SLIM, PRICE-S, SASET, CHECKPOINT and SOFTCOST. Two major thrusts of these AFIT studies have been to: 1) evaluate the ability of the given model to estimate and 2) measure the improvement in estimating accuracy resulting from calibration.

The statistical measurements employed to make these evaluations are those recommended by Conte, Dunsmore, and Shen in their book, Software Engineering Metrics and Models. These three measurements are briefly described in Table 2 below and are also discussed in detail in Chapter III, Methodology.

Table 2. Conte, Dunsmore, and Shen's Criteria

Measurement	Description	Effective Criteria
Mean Magnitude of Relative Error (MMRE)	The average percentage of the absolute difference between the actual effort expended and the predicted effort	MMRE < 0.25
Prediction level within 25% of actual	The percentage of the estimates the model generates that accurately predict effort within 25%	Pred(.25) >0.75
Relative Root Mean Square Error (RRMS)	Represents the mean value of error minimized by the model	RRMS < 0.25

(Conte et al., 1986: 172 - 176)

Results from the AFIT theses are provided in Table 3[2]. The results have been mixed, but show evidence supporting calibration. Calibration of CHECKPOINT by Mertes in 1996, for example, resulted in a model that generates estimates meeting the criteria for estimating accuracy. Interestingly, CHECKPOINT is the only model that uses function points as a primary measure of size. Specifically, Mertes calibrated using function point data in three categories, as indicated in the table. Each category demonstrated remarkable improvement with calibration. Other AFIT calibration efforts have resulted in moderate improvement, but none considered so groundbreaking.

[2] This table appears in Bernheisel, 1997 and Shrum, 1997 as it reflects the collaborative effort of these researchers.

Table 3. Summary Results from the AFIT Calibration Studies

Author (Year)	Cost Model	Application Type	Cali-bration	Vali-dation	Default Accuracy			Validated Accuracy		
					MMRE	RRMS	Pred (0.25)	MMRE	RRMS	Pred (0.25)
Galonsky (95)	PRICE-S	Mil Ground	X	X	not reported		0.52	not reported		0.48
		Unmanned Space	X	X	not reported		0.36	not reported		0.50
		Missile	X		not reported		0.75	not reported		0.75
		Mil Mobile	X		not reported		0.38	not reported		0.38
Kressin (95)	SLIM	Mil Ground - MIS	X		0.962	n/r	0.00	0.157	n/r	0.83
		Mil Ground - All	X		n/r	n/r	n/r	2.166	n/r	0.08
		Command & Control	X	X	0.621	n/r	0.00	0.666	n/r	0.00
Rathmann (95)	SEER-SEM	Avionics	X	X	0.923	1.472	0.25	0.243	0.240	1.00
		Command & Control	X	X	0.531	1.031	0.31	0.311	0.296	0.29
		Signal Processing		X	1.440	1.082	0.06	2.092	1.610	0.43
		Mil Mobile		X	2.802	3.711	0.11	0.462	0.342	0.25
Vegas (95)	SASET	Mil Ground	X	X	10.04	n/r	0.00	5.820	n/r	0.38
		Unmanned Space	X	X	5.54	n/r	0.23	0.940	n/r	0.00
		Avionics	X	X	1.760	n/r	0.00	0.220	n/r	1.00
		Military Mobile	X	X	5.610	n/r	0.25	3.570	n/r	0.00
Weber (95)	REVIC	Mil Ground	X	X	1.21	1.13	0.00	0.86	0.68	0.50
		Unmanned Space	X	X	0.44	0.62	0.50	0.32	0.34	0.50
Mertes (96)	CHECKPOINT	MIS - COBOL		X	0.542	0.101	0.67	0.018	0.010	1.00
	Function Pt	Mil Mobile - Ada		X	1.384	0.412	0.25	0.192	0.057	0.75
	Function Pt	Avionics		X	0.817	0.685	0.50	0.158	0.111	0.75
	SLOC	Command & Control		X	0.193	0.145	0.50	0.165	0.156	0.50
	SLOC	Signal Processing		X	0.090	0.081	1.00	0.090	0.081	1.00
	SLOC	Unmanned Space		X	0.048	0.050	1.00	0.040	0.055	1.00
	SLOC	Ground (supp. space)		X	0.050	0.058	1.00	0.050	0.058	1.00
	SLOC	COBOL Projects		X	0.050	0.051	1.00	0.049	0.051	1.00
Southwell (96)	SOFTCOST	Mil Ground	X	X	1.895	3.433	0.00	0.519	0.870	0.83
		Signal Processing	X	X	0.430	0.612	0.11	0.282	0.634	0.44
		Unmanned Space	X	X	0.557	1.048	0.20	0.480	0.923	0.20
		Ground (supp. space)	X	X	2.734	3.125	0.13	1.802	1.966	0.20
		Military Mobile	X	X	0.635	0.514	0.20	0.420	0.395	0.40
		Avionics	X	X	0.713	0.758	0.20	0.846	0.568	0.20

Note that some results may be misleading. For example, in Rathmann's SEER-SEM calibration, validation of the avionics category shows that the model meets the three proposed accuracy criteria; however, only one data point was used for validation and that essentially resulted in a 'hit or miss' validation. Before examining these studies in more detail, other efforts in the field are briefly summarized.

Other Calibration Efforts. Numerous other efforts have been taken to calibrate or evaluate the software cost models. These efforts are briefly described below.

In a 1994 thesis for the Naval Postgraduate School, Daryl Shadle tests the theory of Abdel-Hamid that it is inappropriate to use raw historical data to calibrate models

because these raw values disregard project inefficiencies such as initial size underestimation. Shadle uses Abdel Hamid's method of simulation to:

> ...test a proposed strategy which capitalizes on the organization's learning experiences by neutralizing the cost excess caused by the initial undersizing, and that derives a posterior set of normalized effort and schedule estimation benchmarks. (Shadle, 1994: v)

Shadle concludes that this process leads to better cost estimates (Shadle, 1994: v).

A series of calibrations was performed by Management Consulting and Research (MCR), Inc. in 1991. The MCR studies focused on calibrating PRICE-S, SEER-SEM, and SASET. SEER-SEM is the only one of the three models for which calibrated estimating accuracy is provided. In this case, SEER-SEM was accurate within six percent for effort for a portion of the data. For PRICE-S, calibrated productivity factor values are identified. For SASET, revised cost estimating relationships are provided (Apgar et al., 1991).

A 1991 AFIT thesis by Gerald Ourada examined the calibration of REVIC, SASET, SEER-SEM, and COSTMODL. Ourada found that SASET and SEER-SEM were uncalibratable with the available data. Ourada also found that accuracy of all the models was significantly low and that none performed as expected. REVIC, for example, was accurate within 25% only 29% of the time for the validation data set (Ourada, 1991).

In 1989, the IIT Research Institute (IITRI) conducted a study of six software cost models on behalf of the Ada Joint Program Office. The main objective was to examine whether cost estimation of programs using the Ada language could be successfully done with cost models that were non-Ada specific. The study found Softcost-Ada (an Ada-specific model) and SASET (a non Ada-specific model) to be accurate within 30% for 4 of 8 projects. They concluded that non-specific models can be tailored for use in specialty domains (IITRI, 1989: vii-x).

In 1987, Chris Kemerer examined the performance of SLIM, COCOMO, Function Points and ESTIMACS in estimating 15 large completed business data processing projects. In this effort, Kemerer validated Albrecht's Function Point model with the available data. He also found that the models not developed to estimate within business data-processing environments did not perform well thereby showing significant need for calibration. Finally, he found that none of the models sufficiently captured the underlying factors affecting productivity (Kemerer, 1987).

In 1981, Robert Thibodeau performed a study for Rome Air Development Center to study the estimating accuracy of nine software cost estimating models. Thibodeau found that calibrated versions of PRICE-S and SLIM were accurate within 30% for estimating effort (Thibodeau, 1981: C-11 - C-13). In general, Thibodeau concludes that "calibration of model parameters may be as important as model structure in explaining estimating accuracy" (Thibodeau, 1981: 6-6).

Lessons Learned from Efforts to Improve Software Cost Models

Although the above studies have shown that the software cost models are not always accurate, there seems to be a generally consistent pattern of at least some improvement in accuracy resulting from calibration. Two questions emerge from this collection of efforts. First, is model calibration really a worthwhile endeavor? Secondly, considering the mixed results in estimating accuracy, are the various cost models useful tools? The available literature indicates that the answer to both these questions is yes.

Value of Calibration. There is an overwhelming consensus in the body of literature with respect to the value of calibration. The following passages are offered:

> ...it is not the accuracy of the models that is being called into question but that they have been used outside their original environment, probably without sufficient attention to calibration and tuning. (Wellman, 1992: 33)

> Before a model can actually be used, it must be adapted to the environment of its use; it must be calibrated. This tests the fit of a model in an organization and enables the model to be adapted to the characteristics of that organization. (Van Genuchten and Koolen, 1991: 41)

> One conclusion is that models developed in different environments do not work very well uncalibrated, as might be expected. (Kemerer, 1987: 427)

> When estimating a software to be developed one should use a tool calibrated on a set of projects belonging approximately to the same software domain of the software to be developed. (Andolfi et al., 1996: 644)

> A more credible approach is to calibrate your selected models with actual data from developments similar to yours...Only after calibration can the models produce truly credible estimates. (Department of the Air Force, 1996: 10-39)

While there is consensus at a conceptual level of calibration, there are some differences in the execution. Some lessons learned can be culled from the research to date and applied to this calibration of the SAGE model. For instance, one of the strengths of the AFIT calibration and validation research since 1995 has been in the general consistency of approach employed. In each effort, an attempt is made to validate the calibration. As discussed previously, validation is the true test of calibration effectiveness. The AFIT studies have also been generally consistent in applying the criteria used to measure accuracy identified in Table 2, except for a small portion of measures that were not reported. This effort to calibrate SAGE will emphasize the consistent use and reporting of these measures.

On the other hand, this SAGE calibration will diverge from the previous AFIT studies in a significant respect. The intent of diverging is to improve the calibration and measurement process. The change involves the manner in which data within a category are split for calibration and validation. The standing practice is to divide the data and use half for calibrating the model and the other half for validating it. The problem is that no matter how random this process is, the calibration results are biased by the split.

Consider, for example, calibration of a category that contains just four data points. It is possible to draw six different sets of two from these four points for calibration. If the points are labeled 1,2,3, and 4, then the possible sets of two would include 1-2, 1-3, 1-4, 2-3, 2-4, and 3-4. The standing AFIT calibration practice is to report on calibration and validation resulting from just one of these six draws of two. For example, points 1 and 2 would be used for calibration and points 3 and 4 would be used for validation. Using computer processing power, however, model calibration and validation resulting from each of the six draws can be simulated. The answers that result from this process are more robust because the whole range of possible outcomes is reported, not just a fraction. This method will be explained in Chapter III.

A separate issue with respect to calibration execution is whether the person who is calibrating the model has enough insight or training with the model to effectively calibrate it. In a portion of the studies above, a thorough analysis of the model is offered as a precursor to calibration, whereas in others it is not. For example, in the AFIT calibration of SEER-SEM, the only description of the model is a paragraph directly quoted from the user's manual. This is not to imply that a lack of a thorough analysis indicates that the author was unfamiliar with the model; however, for the purpose of this research, a deliberate effort is made to explore the basic estimating equations for SAGE before attempting to calibrate it.

A final issue addressed in Shadle's thesis is that it may be inappropriate to use raw historical data to calibrate models. Shadle's results with using Abdel Hamid's computer simulation to normalize raw historical data reveals some promise. Kemerer, on the other hand has criticized Abdel-Hamid's theory:

> ...they [Abdel-Hamid and Madnick] draw the unsupported conclusion that the accuracy of software cost estimation models cannot be judged by their performance on historical (completed) projects. Clearly, ex post estimates cannot

> influence the project. Therefore, it seems reasonable to evaluate them by this method. (Kemerer, 1987: 428)

Clearly, there appears room for discussion and further research in this area. For the purposes of this effort, normalization of historical data through the use of simulation will not be employed.

Calibration of the models appears in fact to be a valuable exercise, given that cost models are to be used. The question then becomes whether the models should be used at all if they do not provide consistently accurate results even when calibrated.

On the Use of Cost Models. The opening section of this chapter introduces Boehm's summary of software cost estimating techniques and his explanation that algorithmic models are one method of several available. Boehm also discusses the complementary nature of the methods and this theme is echoed by others. The following is a recommendation of Van Genuchten and Koolen based on their experience in this field:

> Our first recommendation is: estimate a proposed project by alternative methods. The methods described in this paper are the expert method, the analogy method, and the use of models. If alternative methods are used, the weak points of one method can be compensated by the strong points of another. If the estimates generated by alternative methods agree, the estimate is accepted. If not, the reasons for the difference must be determined, followed by another series of estimates. (Van Genuchten and Koolen, 1991: 44)

If, in practice, it is expected that these various methods are to be used together, then it may be unfair to isolate these models in research and characterize them as being inaccurate. A compounding challenge in the AFIT research stream in this area is that each researcher is one step removed from the data collection effort. The data in each case have been provided by SMC. While SMC has taken efforts to provide accurate data, as will be discussed in Chapter III, the researcher must, in most cases, accept the data at face value. The researcher is not able to either augment or temper the data with information

that someone more involved in the particular program might be able to do. It is assumed that an ability to have more insight into the data would result in better model performance. It is practical to suggest, therefore, that the criteria proposed by Conte, Dunsmore, and Shen may be too rigid to use in judging the accuracy of these models when these models are being evaluated in such a remote and isolated posture.

In light of the discussion and the research, it seems reasonable to suggest that the models are valid tools to use. The models' power is enhanced when properly calibrated and when used in the proper context with other available estimating methods. Having established this basis, it is now appropriate to examine the SAGE model in more detail.

SAGE Model Description

SAGE Model Evolution. Dr. Jensen, through his company Software Engineering, Inc., first introduced SAGE in 1995. The model's roots, however, date to the 1970's, when Jensen worked with the Hughes Aircraft Company. While there, he developed the Jensen macro-level software development estimation model. According to Suzanne Lucas, in a Hughes Aircraft Company report, Jensen's estimating equations were derived over time "based on the work of P.V. Norden, who developed a life cycle cost model for IBM about 1970, and L.H. Putnam who refined Norden's model for the U.S. Army in 1976" (Lucas, 1991: 1-2). Jensen also incorporated the work of Doty, and calibrated the proposed estimating equations to data from software programs at Hughes and to information provided in Dr. Barry Boehm's book, Software Engineering Economics (Lucas, 1991: 2). The resulting equations formed the basis of the Jensen System 1, 2, and 3 models of 1980, which in turn were a precursor to the SEER-SEM model, introduced in 1989. Jensen worked with Galorath Associates in developing and marketing SEER-SEM until the mid-1990's when he decided to work independently to produce SAGE (Jensen, 1997c).

SAGE Model Theory. Jensen lists people, processes, and tools as the three most important elements of software development. He claims that over the past twenty five years, the emphasis in improving software productivity has centered on process and tools, the two technological elements. On the other hand, little, if any, attention has been given to improve productivity through the people element. Jensen says that there can be a substantial difference in the time and effort required to develop a program between a team of people that is properly motivated and well-managed and a team of people that is not. The SAGE model captures the significance of this facet through its unique treatment of personnel and management factors (Jensen, 1996a: 11-13).

Jensen theorizes that effective management of software development teams is the missing link in explaining the disappointing improvement in software productivity since 1960. His review indicates that productivity has increased at less than one source line per person month per year during this period. He says, "This growth is disappointing when the languages, systems, and development environments of 1995 are compared to those available in 1960" (Jensen, 1996a: 1). Jensen agrees with DeMarco and Lister's assessment that any technological innovation that claims 100% improvement in software productivity is 'hot air' (Jensen, 1996b: 90). In light of the limited returns that advanced technological methods have generated, Jensen views effective management as the untapped source in resolving chronic software problems.

The importance of effective management has not gone unnoticed in the field. For example, Dr. Barry Boehm says, "Poor management can increase software costs more rapidly than any other factor" (Boehm, 1981: 486). Humphrey suggests that the unique aspects of software engineering dictates that more management discipline be required, not less (Humphrey, 1990: 30). Humphrey also says that the tendency of managers to view software development as a black art causes them not to use their management instincts to solve software problems (Humphrey, 1990: 30). Despite this seeming consensus on the

important role of software management, Jensen says the traditional software cost estimating models have not adequately captured this pivotal, explanatory element (Jensen, 1996a: 12).

The cost analyst who uses SAGE to build a software cost estimate accounts for management through the model's input parameters . For example, the estimator assigns a very high capability rating for programmers and analysts that come from an organization that is highly motivated and organized. An organization falls into this category if it:

1. effectively uses development teams.
2. has a physical environment that supports the team concept.
3. promotes management styles compatible with team approaches.

(Jensen, 1996a: 3)

Traditional models similarly allow the estimator to input ratings for programmers and analysts but define ratings based on "individual ability and experience since cooperation and communication, necessary elements in team formation, are limited" (Jensen, 1996a: 12).

Jensen's focus on the organizational aspect of software development is the genesis of his concept that it is most appropriate to calibrate a cost model to historical organizational performance (Jensen, 1997c). In that vein, calibration of SAGE to data provided by the Electronic Systems Center is proposed for this effort because that data identifies the developing contractor. SAGE is also calibrated to data provided from the Space and Missile Systems Center; however, this database does not identify the developing contractor. In this case, it becomes necessary to treat the categories for which the software is being developed (e.g., space, avionics) as a collective industry and calibrate within those categories.

An argument can be made that calibrating SAGE to past organizational performance implicitly rewards poor past performance because SAGE generates higher

estimates for poorer performers. The assumption in this argument is that cost estimates are performed in isolation. It ignores the fact that other market competitive forces are in operation which in theory forces poorer performing companies to either improve or go out of the business.

SAGE Estimating Equations. Lucas identifies the following as Dr. Jensen's equations for estimating software development effort and schedule (Lucas, 1991: 2):

$$Ed = 0.4 * S^{1.2} * D^{0.4} * C_{te}^{-1.2} \quad (4)$$

$$t_d = S^{0.4} * D^{-0.2} * C_{te}^{-0.4} \quad (5)$$

where:

Ed = development effort (person-years)
t_d = development time (years)
S = effective system size (source lines of code)
D = system complexity or difficulty
C_{te} = effective technology rating

Dr. Jensen confirms that these equations are used in the SAGE model (Jensen, 1997b). In SAGE, development effort in Equation 4 and development time in Equation 5 refer to the activities and time involved in developing a system from after requirements have been established to before integration activities start. Note that SAGE separately computes estimates for requirements and integration in computing a total effort; however, calibration of equations involving development effort are the focus of this research. Equations (4) and (5) can now be analyzed by examining the behavior of the respective dependent variables, namely development effort and time, with respect to changes in each of the independent variables, namely size, complexity, and effective technology.

Behavior of the SAGE Equation for Development Effort. The behavior of Equation 4 for SAGE effort is summarized in Table 4 and explained in the text that follows.

Table 4. Behavior of the SAGE Equation (4) for Development Effort

An increase in	causes development effort (Ed) to	at a(n)
effective system size (S)	increase	increasing rate.
system complexity	decrease	decreasing rate.
effective technology rating (C_{te})	decrease	increasing rate.

Recall that Equation 4 reads as follows:

$$Ed = 0.4 * S^{1.2} * D^{0.4} * C_{te}^{-1.2} \quad (4)$$

Development effort increases at an increasing rate with respect to size or source lines of code. For example, a doubling of the lines of code causes effort to increase by a factor of 2.3, all other equation values held constant. Jensen and Tonies use the thermodynamic concept of entropy to explain this behavior. They say that in nature, energy is dissipated or wasted by "such things as friction and resistance, heat loss, turbulence, random motion and disorder" (Jensen and Tonies, 1979: 49). They equate this to the human realm in which energy is wasted by "uncooperativeness, incoherence, confusion, undirected or misdirected action" (Jensen and Tonies, 1979: 50). They say that as a software program becomes larger and more people are added, the potential for these factors that cause entropy increase substantially (Jensen and Tonies, 1979: 50-53). This phenomenon is shown in SAGE by the exponential value of 1.2 for the independent variable of size and is referred to as the entropy exponent. Many of the parametric cost estimating models capture entropy in a similar fashion. Intermediate COCOMO, for example, uses values of either 1.05, 1.12, or 1.20 depending on the development mode of the program (Boehm, 1981: 117).

Development effort decreases at a decreasing rate as system complexity increases. This behavior does not appear to match the specified equation and seems counterintuitive. A lower complexity value (or D-value) in SAGE, however, means that the program is actually more complex. In SAGE, complexity values can range from 4 to 32. The characteristics associated with various complexity values are provided in Table 5.

Table 5. Characteristics of Various SAGE Complexity Ratings

Complexity (D) Rating Value	Characteristics
4	Development primarily using microcode for application. Signal processing systems with extremely complex interfaces and control logic
8	New Systems with significant interface and interaction requirements with larger system structure. Operating systems and real-time processing with significant logical code
15	New standalone systems developed on firm operating systems, Minimal interface problems with underlying operating system or other system parts
28	Extremely simple software containing primarily straightline code, simple I/O, and using only internal arrays for data storage.

(Jensen, 1996b: 167)

A program with a complexity value of 8 actually requires 1.3 times more effort than a more complex program with a complexity rating of 4. The rationale for SAGE's model behavior is that more complex programs require more structured design practices. In reaction, managers tend to put more accomplished designers on complex programs and tend less to add people later who will take a long time to effectively contribute to the program's development. The tradeoff is that the program will take a longer time to complete.

Finally, development effort decreases at an increasing rate as the effective technology rating (Cte) increases. In SAGE, higher scores for this constant typically indicate that the organization has better management in place or that the organization is

developing a system that has less constraints placed on it. The details of this constant will be explained later in this chapter. Notice that the exponent for Cte (i.e., -1.2) is the same as the exponent for size, except for that it is negative. Recall that the SAGE exponent for size captures the phenomenon of entropy and the fact that as size grows, more disorder occurs. Therefore, the negative exponent for Cte can be seen as capturing those factors and elements in development that help contain this disorder.

Behavior of the SAGE Equation for Development Schedule. The behavior of Equation 5 for SAGE effort is summarized in Table 6 below and explained in the text that follows.

Table 6. Behavior of the SAGE Equation (5) for Development Schedule

An increase in	causes development schedule (td) to	at a(n)
effective system size (S)	increase	decreasing rate.
system complexity	increase	decreasing rate.
effective technology rating (C_{te})	decrease	decreasing rate.

Recall that Equation 5 reads as follows:

$$t_d = S^{0.4} * D^{-0.2} * C_{te}^{-0.4} \qquad (5)$$

It is relevant to examine the behavior of Equation 5 for schedule in the context of a system of equations that interacts the behavior of Equation 4 for effort. For example, as effective system size increases, the development schedule increases, but at a decreasing rate. Recall that for Equation 4, development effort increases at an increasing rate with respect to system size growth. The interaction of the equations, therefore, is that while both effort and schedule increase as size grows, an increasing rate change in effort is 'countered' by a decreasing rate change in schedule. As in Equation (4), the benefits of having an increased effective technology rating help handle the challenges brought about by increased size, as reflected by the exponents that match for size and effective

technology except in sign. Finally, schedule increases at a decreasing rate as complexity increases. Recall from Equation 4 that in SAGE, more complex programs actually require less effort. The trade-off here is that it takes longer to develop more complex programs. A final note with respect to effort and schedule is that Jensen uses a Rayleigh distribution curve to represent the manner in which effort is allocated over the available schedule.

SAGE Effective and Basic Technology Ratings. The effective technology rating was discussed above in general terms; however, a closer examination provides valuable insight. Lucas notes that Jensen's technology rating:

> ...reflects the developer's software implementation capabilities in the proposed development environment, including personnel capabilities and experience, development support environment, product development requirements, development environment complexity, and target environment. (Lucas, 1991: 2)

The effective technology rating (Cte) is actually a composite score resulting from 31 inputs that the analyst enters into the model. Seven of these inputs factor into a basic technology rating (Ctb) that typically reflect elements that are unique to the organizational environment such as programmer and analyst capabilities. The other 24 factors typically pertain more to the constraints imposed by the system. The effective technology rating is computed from the basic technology rating by the following equation (Apgar et al., 1991: III-2). (Note: Apgar et al. identified this equation in their discussion of SEER-SEM. This researcher has found upon using SAGE, that the same equation holds.)

$$C_{te} = C_{tb} / \prod_{i=1}^{24} f_i \qquad (6)$$

where:

C_{te} = effective technology rating

C_{tb} = basic technology rating, computed from 7 input factors

f_i = SAGE input factors, other than those used to compute C_{tb}

The input factors can assume a range of values around a nominal score of one for each input. In fact, SAGE allows the user to enter a worst case, best case, and most likely value for each input. SAGE in turn produces a range of estimates based on the range of inputs provided.) Table 7 reflects a range of possible scores for the basic technology rating. The first row reflects the highest score, that is when all of the input values are at their highest ratings. This score reflects an organization that has an optimal management approach, employing the best people. The second row reflects the nominal rating, that is when all input values are entered at the nominal score of one. The third row reflects the lowest possible score. The author obtained these values by merely entering the range of values into SAGE for those inputs used to compute Ctb. The final row labeled "Aerospace High" reflects what Dr. Jensen's research has found to be the highest value obtained for a software development environment in the aerospace industry (Jensen, 1996a: 8).

Table 7. Range of SAGE Basic Technology Rating Values

Rating	Basic Technology Rating Value
Highest	29,487
Nominal	5,707
Lowest	1,847
Aerospace High	8,635

Calibration Methods for the SAGE Model. Considering the effort and schedule equations presented above, SAGE model calibration can occur in one of two ways. First, it is possible to simultaneously calibrate Equations 4 and 5 for both effort and schedule. This requires that actual effort, actual schedule, and effective size for each available historical program be entered into the two equations. Calibration would then involve simultaneously solving Equations 4 and 5 for system complexity (D) and the

effective technology rating (C_{te}). This is the method in which the SAGE "point calibrate" function works. The second method is to calibrate only Equation 4 for development effort. This involves entering actual effort, actual size, and complexity into Equation 4 and then solving the equation for the technology rating.

As will be detailed in Chapter III, the second method, calibration of the effort equation only, will be employed in this thesis. The first method is impractical because of the relatively small number of available data points. Boehm, as discussed earlier, recommends that at least ten points be available for a stable simultaneous calibration. Based on previous AFIT calibration efforts, it is uncommon to obtain more than ten data points once an initial stratification of the data is accomplished. Furthermore, only a portion of these data points actually contain information about schedule. Rathmann, in his calibration of SEER-SEM, indicates that he did not calibrate according to schedule because in most of the projects, schedule was not known (Rathmann, 1995: 24).

Summary

This chapter has examined the nature of software cost estimation and describes algorithmic models as one method of several available to estimators. In response to the first research objective proposed in Chapter I, a critical review of efforts taken to improve the estimating accuracy of these models is offered. The review reveals that model calibration is considered a viable means to improve estimating accuracy. A new technique of using simulation to assist in model calibration and validation is presented for further expansion in the following chapter on methodology. Finally, an analysis of the SAGE model and its basic estimating equations are presented as a foundation for establishing the SAGE calibration methodology.

III. Methodology

Introduction

This chapter establishes the method for calibrating and validating the SAGE model to data provided from the Electronics Systems Center (ESC) and the Space and Missile Systems Center (SMC). Initially, all of the steps are introduced and then each step is explained in detail. Through this, insight is gained into: 1) the data's nature; 2) the exact procedures for calibrating and validating the model; and 3) the measures used to evaluate the model's predictive ability.

Overview of Steps

The steps involved in calibration and validation of this model are summarized in Figure 1 below.

1. Collect the data.
2. Stratify data into pre-defined categories.
3. Refine the data by eliminating inappropriate projects and normalizing the data as appropriate. After refinement, determine whether sufficient data remains within category to continue calibration and validation.
4. Enter data points into the SAGE model. Run model (uncalibrated mode) to generate estimates for each project in a category.
5. Calibrate model for each project in the category according to specified procedures.
6. Use a computer simulation model to perform all possible combinations of calibration and validation for the given category of data.
7. Use statistical metrics to establish calibrated model accuracy and to measure improvements resulting from calibration.

Figure 1. Overview of Steps in SAGE Model Calibration

Data Collection (Step 1)

Data for this effort are provided from two sources: the Space and Missile Systems Center (SMC) and the Electronic Systems Center (ESC).

SMC Database. The SMC database is a PC-based compendium of 2,638 historical records of software programs that contain a composite of 50 million source lines of code. Each historical record is characterized by up to 276 parameters which match the input structures of many cost models. Records from both development and maintenance programs are included. SMC considers the database proprietary, and prospective users must seek permission from SMC/FMC (Stukes and Patterson, 1996).

The database dates to 1983 when SMC started to collect historical records of software programs. In 1988, SMC decided to expand their data collection efforts by joining forces with the Space Systems Cost Analysis Group (SSCAG). SSCAG is a cooperative effort between industry and government that meets several times each year to discuss current space-related topics. The SSCAG has a software subgroup that oversees the management of database and provides a non-proprietary version of the database to any SSCAG member organization that contributes data to the collection effort (SSCAG, 1995: E-1). Sources of data include SMC, the European Space Agency, NASA, and major aerospace companies. MCR Federal, Inc., administers the database and indicates that the database has been used for program cost estimates, model calibration, and additional research efforts (Stukes and Patterson, 1996).

The SMC database has both strengths and limits. Its strengths include its volume of data, data dictionary, standardized format for each historical record, and user-friendly query system. Since each contributor is anonymous, there is likely less chance that contributors will distort the data to portray themselves in a more positive light. The database managers also re-contact the sources to verify the inputs (Stukes and Patterson, 1996). On the other hand, the anonymity of the sources poses a limit in that calibration

cannot be performed for each contractor's historical performance. Also, although each record contains up to 276 parameters, a portion of these parameters are not filled in for some programs, leaving gaps in the data. In some records, for example, the effort that it took to develop the program is not provided. This effectively eliminates the possibility of using that record for either effort equation calibration or validation.

ESC Database[3]. The current version of the USAF Electronic Systems Center (ESC) database is a collection of 52 completed software development projects from 32 defense contractors and dates back to 1974. These software projects were developed as either stand-alone efforts or to be integrated with military systems procured by ESC at Hanscom Air Force Base. Typical ESC systems involve high-technology communications and signal processing functions such as those provided by the Airborne Warning and Control System. The data have been collected through a collaborative effort between the developing contractors, ESC and the MITRE Corporation, a government support contractor located near Hanscom. ESC recently updated the database and plans to use it to validate several parametric estimating models including REVIC, PRICE-S, SEER-SEM, and COCOMO 2.0. ESC is also using the database to derive its own software cost estimating relationships (Wells, 1997).

Detailed information for each of these projects is provided at the computer software configuration item (CSCI) level. A CSCI can be considered a mini-project in that each CSCI is developed to perform a distinct end-use function (Ferens, 1997). In all, the 52 project database consists of information for 312 CSCIs. The size of these CSCIs ranges from 411 Executable Deliverable Source Instructions (EDSI) to 448,523 EDSI. The data are provided in a Microsoft© Excel spreadsheet. For each CSCI, over 60 columns of information are provided and include information such as the contractor,

[3] This section was co-authored by Shrum, who is also performing a model calibration using the ESC database.

project effort, CSCI size, schedule, system constraints, and developing contractor capabilities. Since the ESC database was originally constructed for use with SEER-SEM, the information and ratings provided for each CSCI correlate closely with the terminology of SEER-SEM inputs. Since the development contractor is identified for each CSCI, the ESC database is proprietary in nature (Wells, 1997).

Several potential limitations of the data have been identified. These limitations were discussed in a May 1997 telephone interview with Ms. Peggy Wells, the ESC SWDB manager, who provided information regarding ESC's endeavors to remove, or at least reduce these limitations from the database. First, many people, who may have interpreted the exact meaning of each data category in significantly different ways, actually collected the data. Therefore, the data represent a certain degree of subjectivity injected by this collection process. This subjectivity has been mitigated to an extent by a normalization process in which one person has actually entered the date in order to enforce a standardized and consistent assignment of ratings and measures for each CSCI (Wells, 1997). Another limitation was identified by Ourada in his 1991 thesis, in which he indicated that some of the information for each project was incomplete (Ourada, 1991: 4.1). This is still true in the current version of the SWDB. Information regarding software application type for each CSCI, for example, is presented for only a small number of projects. ESC is currently updating the SWDB to provide complete information for this data category.

A serious concern cited by Ourada was that several of the data points represent projects that were never completed but were estimated for completion. Therefore, these projects do not represent actual size or effort (Ourada, 1991: 4.1). However, according to Peggy Wells, action has been taken to remove projects that were never completed from the database (Wells, 1997).

Data Stratification (Step 2)

For this effort, information from the two databases is handled separately. This essentially enables two distinct calibration and validation efforts to be performed, one for the ESC database and one for the SMC database. The fundamental difference in calibration methodology for these two sources of information can be explained in how the data are stratified. For the SMC database, the data are stratified primarily by program category or operating environment. For the ESC database, the data are stratified primarily by the developing contractor.

SMC Database Stratification. In previous AFIT calibration efforts, the SMC data have consistently been stratified by the environment in which the software operated. These environments include: Military Mobile, Military-Specification Avionics, Military Ground and Application - Command and Control, Military Ground and Application - Signal Processing, Unmanned Space, Ground in Support of Space, and Missile (Mertes, 1996: 29). In order to maintain consistency, the data for this effort will be stratified along the same lines.

Since the SMC database is an automated retrieval system, it is relatively easy to search for the records along the above-identified lines. This is performed by the following steps:

1. Start the SMC database.
2. Go to menu item 'File' and select 'Query'. The main query menu screen will appear.
3. Enter Query Title (e.g., 'Military Mobile').
4. Stratify the data by using the query process. There are five primary criteria along which the data can be stratified. These are:

a. Software level - Double-click "Software Level" in the main query screen and select "CSCI" to indicate that information resulting from the query must be provided at the CSCI level only.

b. Operating environment - Double-click "Operating Environment" in the main query screen and select the environment along which the primary stratification is to be accomplished. For example, select "Military Mobile".

c. Applications - Double-click "Applications" in the main query screen and use the "Select All" option to indicate that query must return information on all applications within the selected operating environment. Note for certain operating environments, there are enough projects such that calibration can be focused on one application within that environment. For example, in the "Military Ground" operating environment, there are enough projects grouped for the applications of "Command and Control" and "Signal Processing" to perform a separate calibration for each.

d. Software Functions - Double-click "Software Functions" in the main query screen and use the "Select All" option to indicate that query must return information on projects of all functions, subject to the other query constraints.

e. Programming Language - Double-click "Programming Language" in the main query screen and use the "Select All" option to indicate that query must return information on projects of all languages, subject to the other query constraints.

5. Limit the query to return information on projects that identify the project's size and effort. This is accomplished in the main query screen by unmarking the box labeled "Empty" to the right of the fields for "Effective Size Range", "Total Size Range", and "Effort Range". The query can be further refined by identifying the ranges of values for these three fields.

6. Double-click "Run" at the bottom of the main query screen. The database will then return all records that match the specified criteria.

ESC Database Stratification. The primary stratification of the ESC database for this effort is by contractor or organization. This is in response to Dr. Jensen's philosophy that calibration is most effectively performed against an organization's historical performance. Since the ESC database is provided on a Microsoft® Excel spreadsheet, stratification is performed by accomplishing a primary sort of the data by contractor.

Data Refinement (Step 3)

Data Elimination. After the initial stratification, it is necessary to eliminate unreasonable or inappropriate data points. For this step, the criteria identified by Apgar et al. in the MCR 1991 calibration report of SEER-SEM are used as a basis. The use of these criteria were deemed appropriate by Jensen for this effort (Jensen, 1997a). The criteria are to eliminate data where:

1. Actual Effort = 0.
2. Actual Size = 0.
3. Data provided at a project level other than the computer software configuration item (CSCI) level.
4. CSCI size is larger than 150,000 lines of code.
5. CSCI size is smaller than 1,500 lines of code. (Apgar et al., 1991: III-11)

The following elimination criteria are also used:

6. Projects are developed in a country other than the United States. Development practices in other countries may differ and thus lead to anomalies in the historical data.
7. For the ESC database, contractors whose work was limited to one project are eliminated. The desire is to assess contractor performance over a cross-section of efforts.

Data Normalization. Data normalization is required in some cases to match the format of the provided data to the format assumed in the model. Recall that the SAGE

equation for development effort includes the development activities after the requirements have been defined and before integration activities occur. SAGE also assumes that the basis for a person-month of effort is 152 hours. For each project, the SMC database identifies a figure for "normalized effort" which essentially includes both these same activities and the same 152-hour basis. The SMC figures for effort can therefore be used without further normalization. Similarly, the SMC figures for "effective size" are used as a direct input into SAGE, since there is insufficient information in the database to compute effective size as defined by Jensen in the SAGE model.

On the other hand, the effort identified in the ESC database requires some normalization. The ESC database provides information for total effort, with and without integration. When it is provided, the information for total effort without integration is used as a direct input to SAGE for development effort. In some cases, however, total effort with integration is the only information provided. It becomes necessary to discount integration from this total effort by using the figures that Kressin says are used in the SMC database to normalize effort (Kressin, 1995: 41).[4] To discount integration, total effort is reduced by 7.2 percent. With respect to software size, the ESC database provides information on "equivalent delivered source instructions" which is used directly as the figure for effective size for SAGE.

Calibration and Validation Determination. After the initial refinement of the data, if less than four projects within a category remain, then that category is eliminated from further consideration. If the category contains four or more projects, the category is retained for calibration and validation. For the retained categories, all but two of the projects are used for calibration; the two projects that are held out are used to validate the

[4] The author confirmed the use of these normalization percentages upon examination of the data provided in the SMC database.

calibration. This is a departure from the standing AFIT research practice in which categories with less than eight projects are eliminated and the projects with more than eight are split in half for calibration and validation. The standing practice is motivated by the desire to achieve a degree of stability in the calibration and validation process.

Computer simulation as used in this effort, however, induces stability and also enables greater flexibility. The simulation allows calibration and validation to be performed on each possible combination of data points that can be pulled from a category of projects. For example, if a category has ten projects, then there are 45 combinations in which the ten projects can be split into subsets of eight for calibration and two for validation. Simulation makes it possible to consider all 45 combinations, instead of the traditional method of using just one combination. The robust nature of this process makes it possible to consider calibrating with less than eight projects and also to use a greater proportion of the data to derive a calibration factor.

Enter Data into Model and Generate Estimates (Step 4)

This step involves entering the data for each project into the SAGE model. Before data can be entered, however, it is necessary to create a starting point of input values in the model that is common for all projects. In SAGE, templates can be used to provide such a point. SAGE contains pre-defined templates that contain values for characteristics of an organization that might be considered "common", "traditional" or "modern". The template for modern organizations, for example, assigns higher starting capability ratings for programmers than the template for the traditional organizations does. These starting values are then either adjusted as detailed project information is made available or remain "as is" in the absence of this information. The pre-defined templates are not used in this effort since the ESC and SMC databases provide little, if any, insight into the developing organization's overall approach. A "nominal" template

is therefore created and used for all projects. The nominal template assumes that the value for each model input is assigned a rating of 1.00, except for those values identified in Table 8 that follows. Each assessment of a nominal rating was made by comparing the definitions for a nominal rating provided in the SMC SWDB User's Manual data dictionary to the numerical rating system of inputs employed in the SAGE model.

Table 8. SAGE Nominal Template Values (Other than One)

Model Input (Abbreviation)	Nominal Rating
System Complexity (D)	15
Development System Experience (DEXP)	1.04
Development Methods Experience (PEXP)	0.93
Target System Experience (TEXP)	1.02
Development System Volatility (DVOL)	1.07
Quality Assurance Level (QUAL)	1.03
Specification Level (SPEC)	1.12
Test Level (TEST)	1.05

The nominal template is thus formed by entering all nominal ratings into the model and then saving these values as a product template using the template feature in the SAGE model main menu. Use the SAGE menu sequence of steps, "Template" "Product", and "Add" to save the nominal template. Note that the basic and effective technology ratings associated with this nominal template are 5707.3 and 4463.6, respectively.

Once the nominal template is formed, the SAGE model is ready to accept project inputs from the database. First, the nominal template is opened by using the menu sequence of steps, "Template", "Product", and "Open". Next, the user enters all available information about a project into the model through the various SAGE input screens. If no information is available for a model input, then the input remains at its nominal value. Care must be taken to enter the correct input values from the database into SAGE. Both the ESC and SMC databases use word identifiers to define an input, whereas SAGE uses a number identifier. For example, the SMC database uses the rating of 'very high' under

the category "Display Requirements" to indicate that a software system has a requirement for an interactive display with a light pen, mouse, and touch screen (SMC SWDB User's Manual, 1993: B-13). SAGE, on the other hand, uses the numerical rating of 1.11 for the same definition of display requirements. Appendix A provides a matrix that correlates SMC SWDB and SAGE input values and Appendix B provides the same for the ESC database. The matrices were derived by comparing the ratings used in the model with those used in the databases. After all information for a project is entered, the information for a project is saved as a task using the "Task" and "Save As" menu sequence in SAGE.

The above step is repeated for each project in which information is known. For ease in later use, it is recommended that a subdirectory be established to keep together project files for a like operating environment, application area, or contractor.

Calibrate SAGE Model for Each Project (Step 5)

In this step, a calibrated Ctb value is derived for each project that was entered into SAGE in the preceding step. As discussed in Chapter II, this research involves calibration of the SAGE model equation for development effort only. A simultaneous calibration of both the effort and schedule equations was considered, but determined inappropriate because of the lack of data points after the initial stratification by category was made.

Specifically, the calibration will focus on Equation 4 that was introduced in Chapter II. Recall:

$$Ed = 0.4 * S^{1.2} * D^{0.4} * C_{te}^{-1.2} \qquad (4)$$

Also recall from Chapter II that calibration can be done in one of two ways. The first is to modify the equation weights. For example, the current exponent for size (1.2) can be mathematically computed to be some other number that better reflects the history of cost growth with respect to size for a given project category. The second method, which is

used in this thesis, is to compute an average value for one of the independent variables and then use that value as a constant for future estimation. In Equation 4, the variable that will be calibrated is actually a factor within the effective technology rating (Cte) equation, namely the basic technology rating (Ctb). First, recall Equation 6 for C_{te}:

$$C_{te} = \frac{C_{tb}}{\prod_{i=1}^{24} f_i} \qquad (6)$$

The effective technology rating (Cte) captures the impact on the program resulting from: 1) system characteristics, other than that already accounted for by size (S) and complexity (D), and 2) developing organization characteristics and environment, such as its experience and management approach . The basic technology rating (Ctb) specifically accounts for seven key organizational factors, identified in the following table.

Table 9. Inputs to the Basic Technology Rating (Ctb)

Factor	SAGE Model Abbreviation
Analyst Capability	ACAP
Application Experience	AEXP
Programmer Capability	PCAP
Modern Practices Use	MODP
Automated Tool Support	TOOL
Terminal Response Time	RESP
Hardcopy turnaround time	TURN

By calibrating the basic technology rating, average performance and capabilities for either an organization (in the ESC calibration) or for a class of projects (in the SMC calibration) is derived. Note that the equation used to compute Ctb is proprietary, which makes isolating SAGE calibration to a subset of the seven inputs impossible. The mechanics of using the SAGE model and equations to perform this calibration are now explained.

The crux of the calibration is to find the basic technology rating (Ctb) that equates estimated effort with actual effort for a given project. This is accomplished through

algebra and is made easier with the aid of a computer spreadsheet. (Note: Although the SAGE model has a "Point Calibrate" function in its tools menu, it is not used since it employs the previously described method of simultaneously calibrating effort and schedule. Insufficient data are available to perform a stable simultaneous calibration). The calibration steps are listed below:

1. Equation 4, as previously stated, identifies effort in person-years whereas the SAGE model estimates are in person-months. For calibration, it is necessary to rewrite Equation 4 in terms of person-months:

$$Ed = 4.722 * S^{1.2} * D^{0.4} * C_{te}^{-1.2} \quad (4a)$$

where Ed is the development effort in person-months and all other variables are as previously stated. Note that the author computed the 4.722 coefficient by comparing several inputs to and outputs from the SAGE model. Dr Jensen has confirmed that the coefficient is correct (Jensen, 1997b).

2. Next, Equation 4a is solved for Cte by the following process:

   a. Take the natural logs of both sides of the equation.

   $$\ln(Ed) = \ln(4.722) + 1.2 * \ln(S) + 0.4 * \ln(D) + -1.2 * \ln(C_{te})$$

   b. Isolate the term that contains C_{te}.

   $$1.2 * \ln(C_{te}) = \ln(4.722) - \ln(E_d) + 1.2 * \ln(S) + 0.4 * \ln(D)$$

   c. Divide through by 1.2.

   $$\ln(C_{te}) = \frac{1}{1.2} * \ln(4.722) - \frac{1}{1.2}\ln(E_d) + \ln(S) + \frac{1}{3} * \ln(D)$$

   d. Take the exponents of both sides of the equation.

   $$C_{te} = 4.722^{\frac{1}{1.2}} * E_d^{\frac{-1}{1.2}} * S * D^{\frac{1}{3}} \quad (7)$$

3. Since the proposed method is to calibrate C_{tb}, next substitute Equation 6 into Equation 7 and solve for C_{tb}:

$$\text{a.}\quad \frac{C_{tb}}{\prod_{i=1}^{24} f_i} = 4.722^{\frac{1}{1.2}} * E_d^{\frac{-1}{1.2}} * S * D^{\frac{1}{3}}$$

$$\text{b.}\quad C_{tb} = \prod_{i=1}^{24} f_i * 4.722^{\frac{1}{1.2}} * E_d^{\frac{-1}{1.2}} * S * D^{\frac{1}{3}} \qquad (8)$$

4. For a project, the calibration factor is the basic technology rating (Ctb) that results from inserting known, historical values into the right hand-side of Equation 8. A spreadsheet can be used to automate this process for a collection of programs. The portion of a spreadsheet used in this effort is provided in Figure 2 and is used for further explanation.

Col. 1	Col. 2	Col. 3	Col. 4	Col. 5	Col. 6	Col. 7	Col. 8	Col. 9
	Size (S)	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte (fi product multiplier)	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Prjct 1	100000	12.00	2035.40	4575.10	2.25	1365.90	2000.00	3,329.59

Figure 2. Sample Spreadsheet for SAGE Calibration

In Figure 2, entries for spreadsheet columns 2 through 5 are those values taken from SAGE after all project information has been entered into the model. Column 6 represents the value for $\prod_{i=1}^{24} f_i$ that results from dividing the value in column 5 by the value in column 4. Column 7 is the estimate of development effort generated by SAGE. Column 8 is the actual historical development effort required for this project. The calibrated basic technology rating in column 9 is computed by inserting the values from columns 2 (size), 3 (complexity), 6 (product multiplier), and 8 (actual effort) into equation 8. Note that in this example, the calibrated basic technology value (3329.59) is lower than the model's original estimate (4575.10) to reflect the longer time than estimated that it took to develop this project.

This step identifies the derivation of the calibration factor for a single project. A spreadsheet, similar to the one in Figure 2, is built for each group of projects by entering the values generated by the preceding steps for each project of a calibration category into the spreadsheet. The output of this step is a spreadsheet for each grouping of projects. Each project has a unique calibration factor associated with it. At this point, a final refinement of the data is made. Those projects that have an exceptionally high or low calibration Ctb rating are eliminated as anomalies. The general guidelines used to make this assessment are provided in Table 7, Range of SAGE Basic Technology Rating Values. Generally, projects that are outside of the range from the lowest possible Ctb score (1,847) to the recorded high for the aerospace industry (8,635) are eliminated. For example, a project in the avionics operating environment is eliminated since its calibration value of 58,000 is almost seven times higher than the aerospace high. This particular project contained over 37,000 lines of code and was forecast to consume as much as 500 person-months of effort, but the recorded actual effort was just 39 person-months. Ideally, instead of eliminating the project, it is preferable to either validate or correct the historical information provided by the source; however, such an effort is not possible within the scope of this effort. Next, calibration and validation of the group of projects using simulation is described.

Use Computer Simulation to Calibrate and Validate Model (Step 6)

In this step, simulation using the Microsoft® Excel and Crystal Ball® software programs is employed to calibrate and validate the model. As discussed in Step 3, the idea of using these two tools is to facilitate calibration and validation on every possible combination of projects that can be pulled from a category. A description of simulation using Crystal Ball® is offered as a precursor to explaining the details of this step.

Computer Simulation using Crystal Ball®. Crystal Ball® is a program that is used to enhance the Microsoft® Excel spreadsheet program. "Through a technique known as Monte Carlo simulation, Crystal Ball® forecasts the entire range of results for a given situation" (Crystal Ball® Manual: 1). The basic idea is that the computer generates a range of outputs for an event by sampling many times from a range of values for inputs to that situation. For example, if a person wants to forecast automobile fuel costs for the next year, the person might define a range of possible fuel prices (e.g., $1.00 to $1.25/gallon) and also forecast gallons to be consumed (e.g., 500 to 600 gallons). In a Monte Carlo simulation step, Crystal Ball® pulls a value from each of these two input ranges and using a programmed spreadsheet equation, multiplies them to forecast a total cost. A range of total costs is generated by repeating this simulation step over and over. The operator specifies the number of simulation steps to be executed. Depending on the host computer's processing power, it is possible to perform thousands of steps within minutes.

According to one text, this process of Monte Carlo simulation or "resampling" is said to date back to World War II when a group of physicists at the Rand Corporation used random-number simulations to study complex processes (University of Maryland, 1997: preface). The text explains that the resampling approach provides an unbiased sample of all possible arrangements and which on average provides an answer that is perfectly sound (University of Maryland, 1997: 2-3 - 2-4). In contrast, the conventional approach would be to determine, with the use of statistics, the precise probability of event occurrences. The authors claim that the resampling method is easier because it does not require the "enormous difficulty of learning how to obtain the right formula" when trying to determine the exact probability of an event (University of Maryland, 1997: 2-3). As an example, the text offers the challenge of statistically determining the probability of drawing five or more spades in a deal of 13 cards and says simulation is a reasonable and

easier alternative to assessing this probability (University of Maryland, 1997: 2-3). Considering the relative ease and accuracy of simulation, its application to calibration and validation is now explained.

Simulation applied to SAGE Calibration and Validation. Simulation, as applied to this research effort, involves sampling from the projects in each category generated by Step 6. In a simulation step, Crystal Ball® and Excel are programmed to take a sample of two projects from a category and mark them as validation data points. By default, the projects that are not selected for validation are used for calibration. For the calibration set, the calibrated basic technology values that were computed for individual projects in Step 6, are averaged to achieve the composite calibration factor for that category. Figure 3, which represents a portion of the spreadsheet for a category of four fictitious projects, is used for illustration.

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8
	Size (S)	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 1	100000	12.00	2035.40	4575.10	2.25	1365.90	5000.00	1,551.58
Project 2	80000	10.00	2360.70	5941.80	2.52	813.18	5000.00	1,307.97
Project 3	95000	8.00	2111.00	5447.60	2.58	1045.31	5000.00	1,478.32
Project 4	50000	9.00	1646.70	5536.80	3.36	683.36	5000.00	1,054.37
Total Prj	4.00							
					Use for Validation?	Calibrate=0; Validate = 1	Calibration Ctb	
Distro 1	1478.32	Val Pt #1		Project 1	TRUE	1	1181.17	
Distro 2	1551.58	Val Pt #2		Project 2	FALSE	0	1181.17	
Cal Value	1181.17	Cal Value		Project 3	TRUE	1	1181.17	
	1181.17	Avg Ctb		Project 4	FALSE	0	1181.17	
# for Vali	2			Sum		2		

Figure 3. Sample Spreadsheet for Simulation

In the spreadsheet, notice that after project information has been entered, two input ranges or distributions are defined using the Crystal Ball® program. Specifically, "Distro 1" and "Distro 2" are discrete distributions defined using the Crystal Ball® custom distribution feature. These distributions each have four defined points; these

points are the four calibrated Ctb scores from Column 8. The distributions are established such that each of the four points has an equally likely chance of being selected during a simulation step. If the point is selected during a simulation run, then it is marked as a validation point. The calibration value, "Cal Value" in the figure, is simply the average of the basic technology values from the projects that were not selected for validation. The spreadsheet is programmed such that if the same point is selected from each distribution during a simulation run, then it is considered invalid, because two distinct points are required for validation. In the above figure, projects 1 and 3 were selected for validation and the composite calibration factor represents the average of calibration scores from projects 2 and 4. This composite calibration factor is then fed into the spreadsheet to compute calibrated estimates for both the set of the projects that was selected for calibration and the set selected for validation.

If this simulation step is repeated enough, it is reasonable to expect each possible pairing of two projects in a category will be selected for validation. In fact, the rule of thumb will be to conduct 100 simulated trials for each possible pairing of data to assure that all possibilities are properly considered. This is reflected in the following table.

Table 10. Simulation Trials Required for Calibration and Validation

# of projects in category	Projects used for calibration/validation	Possible Combinations*	Simulation Trials
4	2/2	6	600
5	3/2	10	1,000
6	4/2	15	1,500
7	5/2	21	2,100
8	6/2	28	2,800
9	7/2	36	3,600
10	8/2	45	4,500
11	9/2	55	5,500
12	10/2	66	6,600
13	11/2	78	7,800
14	12/2	91	9,100
Greater than 14	-	-	10,000

(*'Possible combinations' refers to the number of ways that two projects can be selected from the total number of projects and set aside for validation.)

In fact, the simulation process as applied to this effort may be considered somewhat excessive considering the small number of projects that are expected within each category and therefore the relatively small number of ways that the data can be split for calibration and validation. An argument can be made that algorithms be programmed such that each unique combination is considered once and only once, bypassing the need for a random simulation process; however, the effort to write such a program is considered more than trivial and thus outside of the scope of this research effort. The critical element, regardless of whether simulation or a more precise programmed method is used, is that all possible divisions of data for calibration and validation are considered, not just a small fraction as has previously been the case.

The objective of each simulation step is to enable the comparison of actual development effort, default (i.e., non-calibrated) model estimates of effort, and calibrated model estimates of effort. For each simulation step, a set of comparative metrics must be generated since the projects in the calibration and validation sets change with each sampling. The statistics used to make these comparisons were identified briefly in Chapter II and are explained in detail in the following section.

Measure Model Accuracy and Improvements from Calibration (Step 7)

The final step is to measure model accuracy and to evaluate improvements resulting from calibration. This step is actually accomplished as part of the simulation program described in the preceding step, but is discussed separately since it represents the final major phase of the calibration. The three measures of accuracy that have been proposed by Conte, Dunsmore, and Shen and used consistently in the AFIT calibration efforts since 1995, are continued for this effort. The criteria of mean magnitude of

relative error, relative root mean square, and prediction accuracy intervals were introduced in Chapter II, Table 2 and explained here.

Mean Magnitude of Relative Error. The mean magnitude of relative error (MMRE) actually represents the average estimating error for a group of individual projects. The estimating error for an individual project is called the magnitude of relative error (MRE) and simply involves evaluating the difference between the estimate and the actual value according to the following formula.

$$MRE = \left| \frac{E - \hat{E}}{E} \right| \tag{9}$$

where:

E = actual effort expended.

$\hat{E}$ = predicted effort.

The concept of taking the absolute value of the difference comes into play when computing the MMRE. If absolute value is not considered for individual projects, then it is possible that, when computing a group average error, that overestimates and underestimates tend to cancel each other out, giving the impression of accuracy when in fact there may not be. For example, the average error of a 50% overestimate and a 50% underestimate would be zero if absolute value is not figured in. That said, the equation for MMRE is:

$$MMRE = \frac{1}{n} \sum_{i=1}^{n} MRE_i \tag{10}$$

where:

MRE_i = the magnitude of relative error for one project in the group

n = the total number of projects within the group

Conte, Dunsmore, and Shen consider a MMRE value of less then 25% as acceptable for effort production models (Conte, Dunsmore, and Shen: 172).

Relative Root Mean Square Error. The relative root mean square error (RRMS) measures the mean value of error minimized by the model. The RRMS is computed in a three-step process. The first step is to find the mean squared error (MSE) for the group. This involves squaring the raw error for each project and then summing and averaging these squared errors for all of the projects. Since the MSE represents an average squared error, the next step is to take the square root of the MSE to arrive at the root mean square error (RMS). Since this error is in raw terms, the final step is to normalize it so the error can be compared in relative terms to other groups of errors. The result of the normalization is the RRMS. The equations for these three steps are as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(E_i - \hat{E}_i)^2 \tag{11}$$

$$RMS = MSE^{1/2} \tag{12}$$

$$RRMS = \frac{RMS}{\frac{1}{n}\sum_{i=1}^{n}E_i} \tag{13}$$

where:

Ei = actual effort expended for ith project.

$\hat{E}i$ = predicted effort for ith project.

Conte, Dunsmore, and Shen consider a RRMS value of less then 25% as acceptable for effort production models (Conte, Dunsmore, and Shen: 173-175).

Prediction at Level *l* . This measure indicates what percentage of the set of projects meets a target interval for accuracy. For example, if the target interval is 25%, then this measure indicates what percentage of the group's projects have been correctly predicted within a 25% range. The formula for this measure is:

$$\text{Pred}(l) = \frac{k}{n} \tag{14}$$

where:

l = the desired accuracy interval

k = subset of projects within group that are accurate to desired accuracy interval

n = the total number of projects in group

Conte, Dunsmore, and Shen consider an adequate model to be accurate within 25%, 75% of the time (Conte, Dunsmore, and Shen: 173).

The Excel and Crystal Ball® programs track these statistics for each simulation step. Four sections have been partitioned on the spreadsheet to accomplish this. Two of these sections measure default and calibrated model accuracy for the selected calibration projects and the other two sections measure model accuracy for the selected validation projects. The following table provides structure for these comparisons.

Table 11. Calibration and Validation Measurement Comparisons

Area	For projects selected for calibration:	For projects selected for validation:
Default Model Accuracy	Compare actuals to default estimate	Compare actuals to default estimate
Calibrated Model Accuracy	Compare actuals to calibrated estimate	Compare actuals to calibrated estimate

For each quadrant in Table 11, the three accuracy metrics that were just described are generated, such that a total of twelve accuracy metrics are reported. These statistics

represent a large portion of the simulation's output and are identified in Crystal Ball® as forecast cells. Since hundreds of simulation runs are made, a range of scores for each forecast cell emerges. Figure 4 that follows shows the range of MMRE scores generated from a simulated run for two projects selected for validation.

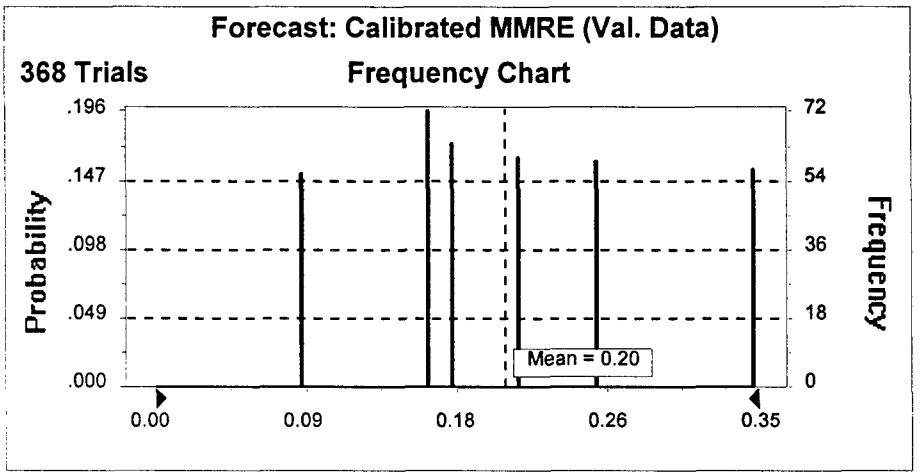

Figure 4. Sample Chart from Simulation Run of Calibration

In this example, the range of MMRE scores represents the calibrated model's accuracy for two validation projects. Note that six lines have emerged and represent the measurements resulting from the six ways that two projects can be sampled from a category of four. Also note that each MMRE score peaks at a different frequency value. In theory, each score should have an equal chance of occurring; however, in simulation, minor random differences can occur. The prior method would result in selection of only one point on this range and depending on the point taken, the model might be construed as being either very accurate (i.e., MMRE value of less than 0.09) or relatively inaccurate (i.e., MMRE value of 0.35). Now, through simulation, a complete range of possible solutions clearly emerges which subsequently enables a better assessment of model accuracy.

A final issue that must be addressed regards deciding which composite calibration factor to recommend to the users of this model. The two viable options are to either use: 1) the mean composite calibration factor that results from the simulation, or 2) the composite calibration factor that results in the lowest estimating error for any two projects selected for validation. The author recommends the first option since it reflects the blending of all possible calibration values in the category as opposed to the one value that happens to minimize error for that set. In other words, the mean score appears to be a more stable reflection of the calibration and validation process.

Limitations

The following items are considered limitations of the methodology provided in this chapter:

1) Calibration of only the SAGE equation for effort is performed. This process, therefore, does not consider the interplay of effort and schedule. For example, consider the case of Project A that required considerably more effort than Project B within a stratified category. It is possible that the extra effort required for Project A was not driven by the relative lack of effectiveness of Project A's development team, but rather because more effort was added in a desire to complete development sooner. It is noted, however, that both a lack of data and a lack of insight into the dynamics of each project does not allow a stable calibration of both the effort and schedule equations.

2) Since calibration does not follow the method espoused by Dr. Jensen in the SAGE Model's "Point Calibrate" feature, it becomes necessary to perform the calibration using a separate spreadsheet program. In addition, the composite value for the basic technology rating (Ctb) that results from this calibration process cannot be input directly into the SAGE Model. It is possible, however, to directly enter the effective technology rating (Cte) using the SAGE "QuickQuote" feature. The SAGE model

estimator who wants to use the results of this calibration feature must, therefore, either 1) manually compute the effective technology rating by inserting the calibrated Ctb value into Equation 6, or 2) alter the inputs in SAGE that are used to compute Ctb, until the estimator arrives at a Ctb score that is reasonably close to the calibration value that results from this effort.

3) The author's experience with the SAGE model has been limited to an academic environment only, with no benefits of knowledge gained by using the model to perform a "real" estimate. A concerted effort has been made, however, to compensate for this limitation by conveying the underlying precepts of the SAGE estimating equations in Chapter II.

4) The potential limitations with respect to the data in both the ESC and SMC databases have already been discussed. The challenge of using this data is compounded, because the researcher was not involved in the collection process and is not privy to any unusual or mitigating circumstances associated with each project, other than those aspects that may be identified in the respective databases.

5) Some of the inaccuracies in cost model estimates are attributable to the uncertainty associated with initial size estimates. As shown, size is a major facet in predicting effort, and it is not necessarily easy to predict size early in a program. In calibration, the uncertainty factor associated with size is essentially eliminated by using actual (i.e., certain) size values. In theory, this elimination of uncertainty could make the model appear more accurate than perhaps it truly is.

Summary

This chapter establishes the methodology for calibration and validation of the SAGE model equation for development effort in response to the second research objective proposed in Chapter I. Computer simulation using Microsoft® Excel and a

supplementary program, Crystal Ball®, are introduced as a means to enable a comprehensive simulation and calibration process. The three criteria used to measure model accuracy are then explained. The context in which these measures are used to demonstrate calibrated model accuracy and improvements resulting from calibration is described. Finally, limitations of this approach are discussed.

IV. Findings

Overview

This chapter provides the results of the calibration and validation of the SAGE software cost estimating model. The results are summarized in this section and detailed in the sections that follow.

From the SMC database, 70 projects from seven separate categories were used for model calibration and validation. The SMC categories were: Mil-Spec Avionics, Military Ground - Command and Control, Military Ground - Signal Processing, Ground in Support of Space, Military Mobile, Missile, and Unmanned Space. Of the seven categories, SAGE returned the best results for the Military Ground - Command and Control category in which the uncalibrated model met two (MMRE, RRMS) of the three proposed criteria for estimating accuracy. Other results were mixed. Only one category demonstrated across-the-board improvement from calibration while two categories actually became worse with calibration. Of the two categories that got worse, however, one already had a high degree of default estimating accuracy and the other had projects in which no detailed information was available and nominal values had to be assumed for all model inputs.

Table 12 that follows provides the weighted average accuracy for the SMC database calibration and validation. The weighted average was computed by: 1) multiplying the number of projects within a category by the mean score for each of the three accuracy measures used; 2) summing the products of step 1 for each measure, from all categories; 3) dividing these sums from step 2 by 70, the total number of SMC projects evaluated.

Table 12. Weighted Average Results for Calibration and Validation of SMC Data

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.40	0.35	12.5% better	MMRE<0.25
RRMS	0.59	0.56	5.1% better	RRMS<0.25
Pred Interval (25%)	0.37	0.41	10.8% better	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.40	0.43	7.5% worse	MMRE<0.25
RRMS	0.48	0.51	6.3% worse	RRMS<0.25
Pred Interval (25%)	0.37	0.32	13.5% worse	Pred (25%)>0.75
Summary Information				
Number Projects Used	70			

* Represents weighted means of the means generated during simulation of each category

From the ESC database, 40 projects were used for model calibration and validation. The projects represent the work of three contractors, referred to herein as Contractor A, Contractor J, and Contractor R. Of the three contractors, SAGE returned the best results for Contractor R, for which the calibrated model met two of the three criteria for estimating accuracy. Furthermore, the results for Contractor A and Contractor R demonstrated across-the board improvement resulting from calibration. On the other hand, calibration of Contractor J's projects caused the model's estimating accuracy to worsen. Table 13 that follows provides the weighted average accuracy for the ESC database calibration and validation. The weighted average was computed in the same manner described above for the SMC database results, except that 40 total projects are considered, instead of 70.

Table 13. Weighted Average Results for Calibration and Validation of ESC Data

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.38	0.37	2.6% better	MMRE<0.25
RRMS	0.68	0.53	22.1% better	RRMS<0.25
Pred Interval (25%)	0.27	0.22	18.5% worse	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.41	0.41	No Change	MMRE<0.25
RRMS	0.50	0.45	10.0% better	RRMS<0.25
Pred Interval (25%)	0.26	0.27	3.8% better	Pred (25%)>0.75
Summary Information				
Number Projects Used	40			

* Represents weighted means of the means generated during simulation of each category

The remainder of this chapter details the results for each calibration and validation category for the SMC and ESC databases. The results are presented in tabular format and augmented by written analysis. After the results from each category are presented, a collective analysis of the results is offered.

Mil-Spec Avionics (SMC)

A total of nine projects was used for the calibration and validation of the Mil-Spec Avionics category. The projects used are identified in Table 14 and the results are provided in Table 15. Three projects (SMC Record Numbers 67, 2615, and 2618) were eliminated because the calibrated Ctb scores for those three projects fell outside of the observed range of reasonable values identified in Table 7.[5]

The Mil-Spec Avionics Category responded well to calibration, although the calibrated model did not meet the accuracy criteria identified by Conte, Dunsmore, and Shen. Calibration and validation results tend to become more erratic and calibrated model accuracy generally becomes worse as the range of calibrated Ctb values grows. In this category, the calibration values for individual projects ranged from 3482.41 to 9663.10. Recall that the calibration value for the entire category is computed by taking the average of the individual calibration values of the projects selected for calibration. Each time a simulation step is performed, a different subset of projects is selected for calibration, such that a range of calibration values for the entire category is generated. For the group, Appendix D indicates that the categorical calibration values ranged from 6018.44 (10% level) to 7154.01 (90% level), with a mean score of 6545.32. Note that only Project 2617 has an individual calibrated value (i.e., 6513.43) that falls within this range. The calibrated model will tend, therefore, to be very accurate for Project 2617. The calibrated model becomes increasingly inaccurate, however, for projects with individual calibration values significantly outside this range. For example, the calibrated model will return poorer results for Projects 302 and 346. A tighter range of calibration values must exist for the calibrated model to return more accurate results.

[5] The decision to eliminate a project based on its calibrated Ctb score is subjective. The range of values in Table 7 provides a guideline for elimination. In some instances, projects with calibration scores outside this range remain if the value is generally consistent with the other calibration values in the category.

Table 14. Mil-Spec Avionics Projects Used for Calibration and Validation

SMC Record #	Size (S)	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Default Model Estimate	Actual Effort	Calibrated Ctb
Project 10	43207	8.00	2720.10	6073.50	2.23	299.59	370.00	5,093.80
Project 11	32878	15.00	2637.90	6073.50	2.30	287.96	198.00	8,298.52
Project 12	22027	21.00	3072.70	5707.30	1.86	169.64	112.00	8,066.69
Project 13	58153	8.00	2720.10	6073.50	2.23	427.90	752.00	3,796.47
Project 14	22148	8.00	1170.00	6658.60	5.69	369.78	464.00	5,511.06
Project 302	45353	8.00	1234.90	5317.70	4.31	819.09	400.00	9,663.10
Project 346	40000	8.00	2330.10	6176.00	2.65	328.84	654.00	3,482.41
Project 2512	33158	8.00	1920.70	6727.60	3.50	331.07	245.00	8,646.07
Project 2617	18000	12.00	6566.90	8633.60	1.31	42.79	60.00	6,513.43
Total Projects	9							

Table 15. Mil-Spec Avionics, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.44	0.34	22.7% better	MMRE<0.25
RRMS	0.59	0.52	11.9% better	RRMS<0.25
Pred Interval (25%)	0.22	0.35	59.1% better	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.45	0.39	13.3% better	MMRE<0.25
RRMS	0.54	0.52	3.7% better	RRMS<0.25
Pred Interval (25%)	0.21	0.24	14.3% better	Pred (25%)>0.75
Summary Information				
Number Projects Used	9			
Mean Ctb Value	6545.32			

* Represents mean value resulting from the simulation run

Military Ground - Command and Control (SMC)

A total of ten projects was used for the calibration and the validation of the Military Ground - Command and Control category. The projects used are identified in Table 16 and the results are provided in Table 17. One project (SMC record number 2517) was eliminated because its calibrated Ctb value fell outside of the observed range of reasonable values identified in Table 7.

The default model met Conte, Dunsmore, and Shen's standards for estimating accuracy for the MMRE and RRMS criteria, each with a score of 0.23. In addition, the model was accurate within 25%, 70% of the time, just short of the target 75%. The calibrated model was actually less accurate than the default model, due in large part to the wide band of calibration values, which ranged from 2673.19 to 8042.78. The calibrated model performed worse for projects in which the default model was already reasonably accurate. For example, consider that the model's default estimate of 92.95 manmonths of effort for Project 150 was within 8% of the actual effort of 100 manmonths. The model's default estimate was based on a Ctb value of 5707.30. The calibrated model, however, dictates the use of 5912.02 as a calibration value. For Project 150, this higher calibration Ctb value computes to a lower estimate, 89.10 manmonths. While still reasonable, the calibrated model is less accurate than the default model.

The event described above is more likely to occur as the range of calibration values widens, essentially because the category's mean calibration value resulting from the simulation bridges the category's extreme high and low individual project calibration values, but its use fails to help estimate either extreme particularly well. In this category, the use of the calibrated Ctb value (i.e., 5912.02) would result in less accurate estimates than the default model for seven of the ten projects, which helps provides some insight as to why this category fails to improve from calibration.

Table 16. Command and Control Projects Used for Calibration and Validation

	Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8
SMC Record #	Size (S)	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 7	45057	12.00	5776.80	6675.50	1.16	150.07	120.00	8,042.78
Project 9	128200	12.00	6017.10	6675.50	1.11	501.19	517.00	6,504.91
Project 50	144000	12.00	4574.30	6154.40	1.35	800.65	684.00	7,017.39
Project 120	25842	15.00	4463.60	5707.30	1.28	114.74	95.00	6,679.91
Project 124	23881	15.00	4463.60	5707.30	1.28	104.38	139.00	4,495.26
Project 145	18560	15.00	4463.60	5707.30	1.28	77.13	101.00	4,558.87
Project 150	21681	15.00	4463.60	5707.30	1.28	92.95	100.00	5,369.83
Project 152	69772	15.00	4463.60	5707.30	1.28	377.88	286.00	7,198.75
Project 155	8398	15.00	4463.60	5707.30	1.28	29.78	74.00	2,673.19
Project 2510	43437	12.00	5679.90	7490.80	1.32	146.56	172.00	6,555.50
Total Projects	10							

Table 17. Military Ground - Command and Control, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.23	0.25	8.7% worse	MMRE<0.25
RRMS	0.23	0.28	21.7% worse	RRMS<0.25
Pred Interval (25%)	0.70	0.59	15.7% worse	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.23	0.29	26.1% worse	MMRE<0.25
RRMS	0.23	0.30	30.4% worse	RRMS<0.25
Pred Interval (25%)	0.70	0.45	35.7% worse	Pred (25%)>0.75
Summary information				
Number Projects Used	10			
Mean Ctb Value	5912.02			

* Represents mean value resulting from the simulation run

Military Ground - Signal Processing (SMC)

A total of 16 projects was used for the calibration and the validation of the Military Ground - Signal Processing category. The projects used are identified in Table 18 and the results are provided in Table 19. Three projects (SMC record numbers 54, 127, and 140) were eliminated because their calibrated values fell outside the observed range of reasonable values identified in Table 7.

The model does not meet Conte, Dunsmore, and Shen's criteria for estimating accuracy, and it failed to respond to calibration for this category. Refer to the discussion in the preceding section on how a wide range of calibrated Ctb values can cause the calibrated model to perform worse than the default model. In this category, the range of individual project Ctb values (2048.99 to 7826.52) is considerable. A significant contributor to this range is the fact that none of these projects had detailed project information, other than project size and effort. The calibrated Ctb value for a project thus essentially "absorbs" the wide variations in project effort that may have otherwise been explained by project information had it been available. For example, compare Projects 142 and 144. These projects have comparable size (28,782 and 29,802 lines of code respectively) but Project 142 required twice as many manmonths of effort than Project 144 required. While it is conceivable that some of the difference in effort required can be explained by the organizational factors that the Ctb value captures, it is doubtful that all of it can be. In the absence of any other project information, however, the calibrated Ctb values are used to explain all variation in the effort. This tends to induce the wide range of calibrated values, that then leads to the model's unresponsiveness to calibration.

Table 18. Signal Processing Projects Used for Calibration and Validation

SMC Record #	Size	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 126	47965	15.00	4463.60	5707.30	1.28	241.02	165.00	7,826.52
Project 130	71851	15.00	4463.60	5707.30	1.28	391.43	738.00	3,364.60
Project 131	29147	15.00	4463.60	5707.30	1.28	132.57	192.00	4,191.70
Project 132	46595	15.00	4463.60	5707.30	1.28	232.78	278.00	4,922.47
Project 133	123710	15.00	4463.60	5707.30	1.28	751.32	645.00	6,481.16
Project 134	44527	15.00	4463.60	5707.30	1.28	220.44	228.00	5,549.13
Project 135	23787	15.00	4463.60	5707.30	1.28	103.88	264.00	2,623.52
Project 136	12121	15.00	4463.60	5707.30	1.28	46.26	154.00	2,094.85
Project 137	60233	15.00	4463.60	5707.30	1.28	316.77	274.00	6,440.55
Project 138	14389	15.00	4463.60	5707.30	1.28	56.83	190.00	2,087.45
Project 142	28782	15.00	4463.60	5707.30	1.28	130.58	348.00	2,521.66
Project 143	23703	15.00	4463.60	5707.30	1.28	103.44	86.00	6,656.85
Project 144	29802	15.00	4463.60	5707.30	1.28	136.16	145.00	5,415.68
Project 147	31720	15.00	4463.60	5707.30	1.28	146.74	192.00	4,561.73
Project 153	11534	15.00	4463.60	5707.30	1.28	43.58	149.00	2,048.99
Project 154	8965	15.00	4463.60	5707.30	1.28	32.21	109.00	2,066.53
Total Prj	16							

Table 19. Signal Processing, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.39	0.46	17.9% worse	MMRE<0.25
RRMS	0.49	0.56	14.3% worse	RRMS<0.25
Pred Interval (25%)	0.44	0.22	50.0% worse	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.39	0.50	28.2% worse	MMRE<0.25
RRMS	0.43	0.54	25.6% worse	RRMS<0.25
Pred Interval (25%)	0.44	0.20	54.5% worse	Pred (25%)>0.75
Summary information				
Number Projects Used	16			
Mean Ctb Value	4301.68			

* Represents mean value resulting from the simulation run

Ground In Support of Space (SMC)

A total of 14 projects was used for the calibration and the validation of the Ground in Support of Space category. The projects used are identified in Table 20 and the results are provided in Table 21. Eleven projects (SMC record numbers 80, 88, 91, 92, 98, 99, 105, 107, 329, 331, and 332) were eliminated because their calibrated values fell outside the observed range of reasonable values identified in Table 7.

The model demonstrated slight improvement from calibration, but fails to meet Conte, Dunsmore, and Shen's criteria for estimating accuracy. Nonetheless, the calibrated MMRE measure of 0.32 indicates that the model is reasonably accurate for this category. The RRMS value (0.44) is driven higher in this case because this measure gives higher weighting to projects of larger effort. For example, consider the estimating error of two hypothetical projects. One is estimated at 10 manmonths and actually consumes 20 manmonths of effort. The second is estimated at 100 manmonths and consumes 200 manmonths of effort. The MMRE for these two projects is 0.50. The RRMS, on the other hand, is 0.65 because the gross error of the second project is given more weight in the RRMS calculation. In this category, the relative large raw estimating error of projects 75, 90, and 117 drives the RRMS higher relative to the MMRE.

It was recognized that Projects 74-79 and 81-83 in this group represented the Command/Control application of this category. Out of researcher curiosity, a separate calibration and validation effort was performed on this subset of projects and the results are provided in Appendix C. As expected, the model was considerably more accurate and more responsive to calibration, since the range of calibration values for these projects was considerably tighter than that represented by the category's other projects.

Table 20. Ground in Support of Space Projects Used for Calibration and Validation

SMC Record #	Size	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 74	11700	12.00	2303.70	5158.50	2.24	89.68	80.00	5,673.80
Project 75	116800	8.00	1785.80	5158.50	2.89	1637.17	912.00	8,399.87
Project 76	14000	15.00	2166.60	5158.50	2.38	130.91	115.00	5,746.81
Project 77	56200	8.00	1785.80	5158.50	2.89	680.53	523.00	6,424.07
Project 78	48300	15.00	2303.70	5158.50	2.24	537.51	478.00	5,688.40
Project 79	50300	15.00	2303.70	5158.50	2.24	564.33	432.00	6,445.12
Project 81	22900	15.00	2303.70	5158.50	2.24	219.51	164.00	6,577.06
Project 82	16300	15.00	2303.70	5158.50	2.24	145.97	140.00	5,341.30
Project 83	6800	21.00	2303.70	5158.50	2.24	58.49	57.00	5,270.95
Project 90	95000	15.00	4463.60	5707.30	1.28	547.28	1055.00	3,302.89
Project 97	80000	15.00	4463.60	5707.30	1.28	445.30	530.00	4,936.37
Project 106	16300	15.00	4463.60	5707.30	1.28	66.00	206.00	2,210.61
Project 115	13000	15.00	4463.60	5707.30	1.28	50.31	109.00	2,996.65
Project 117	66843	15.00	4463.60	5707.30	1.28	358.93	652.00	3,470.54
Total Projects	14							

Table 21. Ground in Support of Space, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.32	0.29	9.4% better	MMRE<0.25
RRMS	0.61	0.61	No change	RRMS<0.25
Pred Interval (25%)	0.43	0.46	7% better	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.32	0.32	No change	MMRE<0.25
RRMS	0.44	0.44	No change	RRMS<0.25
Pred Interval (25%)	0.43	0.43	No change	Pred (25%)>0.75
Summary information				
Number Projects Used	14			
Mean Ctb Value	5176.07			

* Represents mean value resulting from the simulation run

Military Mobile (SMC)

A total of 10 projects was used for the calibration and validation of the Military Mobile category. The projects used are identified in Table 22 and the results are provided in Table 23. Four projects (SMC record numbers 348, 349, 2456, and 2515) were eliminated because their calibrated values fell outside the observed range of reasonable values identified in Table 7.

The calibration of this category met with mixed success, and did not meet Conte, Dunsmore, and Shen's criteria for estimating accuracy. The model responded better to the data used for calibration than it did to the data used for validation. As seen in Table 23, all three of the accuracy metrics improved for the calibration data but only one of the three metrics improved for the validation data. Since the validation set only includes two projects for each simulation run, it can be more susceptible to erratic results. In this category, seven of the ten projects have calibration values between 2900 and 5600, a reasonably moderate range. On the other hand, projects 303, 347, and 2508 have calibration values of 7409.66, 6931.28, and 8000.43 respectively. When any of these three projects are selected for validation, they tend to drive a larger error because projects with much lower calibration values have been used to predict them. If any of these projects are used to calibrate, however, the impact of the magnitude of their large individual calibration values is lessened because these values are averaged with projects that have lower values.

Finally, as discussed in the previous section, the RRMS is driven higher relative to the MMRE when there are several projects of disproportionate effort that also have relatively large estimating error. In this category, six of the ten projects contained actual effort of 250 months or less. Projects 2502, 2503, 2507, and 2508, however, each have over 600 months of effort, which thus drives the RRMS higher relative to the MMRE.

Table 22. Military Mobile Projects Used for Calibration and Validation

SMC Record #	Size (S)	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 34	17134	15.00	4463.60	5707.30	1.28	70.08	83.00	4,956.49
Project 303	30000	4.00	1446.80	5882.80	4.07	312.61	237.00	7,409.66
Project 347	2311	8.00	611.70	5328.90	8.71	53.47	39.00	6,931.28
Project 2500	1958	12.00	2768.60	4971.30	1.80	8.42	14.00	3,253.98
Project 2502	26239	15.00	1397.70	5317.70	3.80	470.76	633.00	4,154.86
Project 2503	32464	15.00	1349.90	5317.70	3.94	633.70	783.00	4,458.19
Project 2505	7448	15.00	1581.90	5317.70	3.36	89.54	180.00	2,971.62
Project 2506	6317	15.00	1581.90	5317.70	3.36	73.48	152.00	2,901.72
Project 2507	26814	15.00	1046.40	5317.70	5.08	683.85	647.00	5,568.90
Project 2508	58789	15.00	731.00	4680.90	6.40	2697.84	1418.00	8,000.43
Total Prj	10							

Table 23. Military Mobile, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.37	0.36	2.7% better	MMRE<0.25
RRMS	0.89	0.71	20.2% better	RRMS<0.25
Pred Interval (25%)	0.30	0.38	26.7% better	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.37	0.41	10.8% worse	MMRE<0.25
RRMS	0.47	0.52	10.6% worse	RRMS<0.25
Pred Interval (25%)	0.29	0.36	24.1% better	Pred (25%)>0.75
Summary information				
Number Projects Used	10			
Mean Ctb Value	5060.08			

* Represents mean value resulting from the simulation run

Missile (SMC)

A total of 4 projects was used for the calibration and validation of the Missile category. The projects used are identified in Table 24 and the results are provided in Table 25. The calibration values of two of the four projects (SMC record numbers 15, and 36) fell outside the observed range of reasonable values identified in Table 7; however, these projects were not eliminated because they were not outliers in relation to the category as a whole.

This category responded well to calibration but it did not meet the criteria identified by Conte, Dunsmore, and Shen for estimating accuracy. In this case, the calibration values are more homogenous with one exception. Project 16 has a calibration value of 4088.22, which is quite larger than the other three projects. In a simulation step, if Project 16 is used for calibration, its large value is averaged with one of the other three smaller values. The resulting composite calibration value (i.e., the average of the calibration values of the two projects selected for calibration) serves as a reasonable estimator for the two projects held out for validation. If, however, Project 16 is selected for validation, then two projects with substantially lower calibration values are used to estimate it, and the accuracy therefore gets worse. This explains why in this category, the average accuracy of the model for the projects used for validation is markedly worse than for the projects used to calibrate.

Note also that the mean categorical calibration value of 2366.29 is the second lowest score from any of the calibration categories. This is indicative of a rather low development proficiency for this environment. This likely suggests that either the development of missile software is burdened with unique challenges that the model does not capture or that the developing organizations are not proficient in developing these systems. It is difficult, however, to draw conclusions on a basis of only four projects.

Table 24. Missile Projects Used for Calibration and Validation

SMC Record #	Size (S)	Cmplx (D)	Cte (SAG Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 15	8885	12.00	2450.80	5707.30	2.33	59.84	262.00	1,667.35
Project 16	9025	12.00	2450.80	5707.30	2.33	60.98	91.00	4,088.22
Project 27	18933	8.00	1095.20	6121.00	5.59	331.61	1384.00	1,860.96
Project 36	13658	8.00	1982.90	6176.00	3.11	109.92	480.00	1,808.12
Total Projects	4							

Table 25. Missile, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.66	0.26	60.6% better	MMRE<0.25
RRMS	0.88	0.30	65.9% better	RRMS<0.25
Pred Interval (25%)	0.00	0.49	Undefined	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.66	0.67	1.5% worse	MMRE<0.25
RRMS	0.89	0.44	51% better	RRMS<0.25
Pred Interval (25%)	0.00	0.24	Undefined	Pred (25%)>0.75
Summary information				
Number Projects Used	4			
Mean Ctb Value	2366.29			

* Represents mean value resulting from the simulation run

Unmanned Space (SMC)

A total of 7 projects was used for the calibration and validation of the Unmanned Space category. The projects used are identified in Table 26 and the results are provided in Table 27. The calibration values of four of the seven projects (SMC record numbers 2622, 2623, 2624, and 2625) fell outside the observed range of reasonable values identified in Table 7; however, these projects were not eliminated because they were not considered outliers in relation to the category as a whole. In addition, three of the projects (SMC record numbers 2623-2625) had identical values for each input and output. The calibration values for these three projects were altered by one-hundredth of a point so that the computer simulation could distinguish between the three projects

This category responded reasonably well to calibration but did not meet the criteria identified by Conte, Dunsmore, and Shen for estimating accuracy. Better accuracy results were expected considering the reasonably tight range of calibration values for the projects; however, Project 3 caused results to be erratic because of its larger calibration value, relative to the other projects. In fact, Project 3 is similar to the case of Project 16 discussed in detail in the preceding Missile category section. Also note that in Appendix D, the 50% confidence values for the validated MMRE and RRMS in this category are each at 0.30 which is close to the target of 0.25, and lower than the mean values of 0.59 and 0.88 respectively. The mean results are driven higher, primarily by the influence of Project 3. Finally, note that the mean calibration value of 2320.91 is the lowest score from any of the calibration categories. This is not too surprising considering the overwhelming complexity associated with developing software for this environment.

Table 26. Unmanned Space Projects Used for Calibration and Validation

SMC Record #	Size (S)	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 3	80000	15.00	4463.60	5707.30	1.28	445.30	583.00	4559.46
Project 305	12810	15.00	4262.70	6121.00	1.44	52.24	143.00	2,644.69
Project 306	9334	21.00	5469.50	6372.40	1.17	30.31	94.00	2,481.20
Project 2622	19810	15.00	2602.10	4896.00	1.88	159.38	558.00	1,723.23
Project 2623	16759	15.00	4463.60	5707.30	1.28	68.24	305.00	1,638.87
Project 2624	16759	15.00	4463.60	5707.30	1.28	68.24	305.00	1,638.88
Project 2625	16759	15.00	4463.60	5707.30	1.28	68.24	305.00	1,638.89
Total Prj	7							

Table 27. Unmanned Space, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Calibrated Model***	**Change from Calibration**	**Target**
MMRE	0.66	0.43	34.8% better	MMRE<0.25
RRMS	0.69	0.88	27.5% worse	RRMS<0.25
Pred Interval (25%)	0.14	0.52	271.4% better	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.66	0.59	10.6% better	MMRE<0.25
RRMS	0.69	0.88	27.5% worse	RRMS<0.25
Pred Interval (25%)	0.14	0.30	114.3% better	Pred (25%)>0.75
Summary information				
Number Projects Used	7			
Mean Ctb Value	2013.59			

* Represents mean value resulting from the simulation run

Contractor A (ESC)

A total of 17 projects was used for the calibration and validation of Contractor A in the ESC database. The projects used are identified in Table 28 and the results are provided in Table 29. Seven projects (ESC Record Numbers 1.2, 2.1, 32.11, 32.12, 33.1. 33.10, and 33.13 were eliminated because their calibrated values fell outside the observed range of reasonable values identified in Table 7.

The calibrated model performed well in this category, and in fact demonstrated across the board improvement in accuracy. The calibrated model, however, did not meet the accuracy criteria identified by Conte, Dunsmore, and Shen. The calibrated model improved because the mean calibration value of 3901.00 serves as a better estimator than the default model for 15 of the 17 projects. The only projects in which the default model is more accurate are ESC record numbers 2.2 and 33.4. Nonetheless, the model fails to meet the accuracy criteria, mainly because of the relatively wide range of calibration values present.

The wide range of calibration values is demonstrated by the following analysis. Appendix D indicates that the categorical calibration value ranges from 3707.72 (10% level) to 4077.94 (90% level); however, only three projects in this category have individual calibration values that fall within this range (ESC record numbers 1.3, 32.2, and 32.4). As discussed previously, the model will be very accurate for these three projects and increasingly less accurate for projects that have values at the high and low extremes of this range.

Table 28. ESC Contractor A Projects Used for Calibration and Validation

ESC Record #	Size (S)	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 1.1	39207	12.00	2493.70	3572.00	1.43	348.03	210.00	5,441.85
Project 1.3	10973	8.00	3704.40	6413.70	1.73	39.93	73.00	3,879.26
Project 2.2	7653	8.00	4277.20	6413.70	1.50	21.81	22.50	6,248.33
Project 32.1	9815	15.00	3531.80	3936.60	1.11	47.56	77.49	2,620.93
Project 32.2	11563	15.00	3768.00	4724.00	1.25	53.57	70.06	3,777.11
Project 32.3	8501	12.00	3968.80	6383.00	1.61	31.83	82.59	2,883.27
Project 32.4	11104	4.00	5868.00	9587.50	1.63	17.67	51.32	3,943.90
Project 32.5	8741	21.00	2096.70	3696.20	1.76	88.52	58.46	5,222.57
Project 32.10	7709	15.00	5305.50	5707.30	1.08	21.84	66.35	2,260.96
Project 33.4	7076	12.00	5187.20	7656.10	1.48	18.52	18.10	7,805.05
Project 33.5	11333	8.00	5898.80	7138.80	1.21	23.75	83.98	2,491.79
Project 33.8	6556	12.00	5364.50	5909.90	1.10	16.23	55.68	2,115.63
Project 33.11	27676	8.00	4485.20	5212.60	1.16	96.33	150.34	3,597.15
Project 33.14	11239	8.00	3135.00	5212.60	1.66	50.21	102.08	2,885.55
Project 45.1	8127	15.00	4471.50	6014.80	1.35	28.57	110.90	1,942.69
Project 45.2	9961	12.00	5959.20	10166.60	1.71	23.63	58.93	4,748.12
Project 45.3	17018	12.00	4471.50	6014.80	1.35	63.44	91.87	4,417.63
Total Projects	17							

Table 29. ESC Contractor A, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.43	0.36	16.3% better	MMRE<0.25
RRMS	0.71	0.49	31.0% better	RRMS<0.25
Pred Interval (25%)	0.18	0.25	38.9% better	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.48	0.41	14.6% better	MMRE<0.25
RRMS	0.57	0.40	29.8% better	RRMS<0.25
Pred Interval (25%)	0.17	0.31	82.4% better	Pred (25%)>0.75
Summary information				
Number Projects Used	17			
Mean Ctb Value	3901.00			

* Represents mean value resulting from the simulation run

Contractor J (ESC)

A total of 17 projects was used for the calibration and validation of Contractor J in the ESC database. The projects used are identified in Table 30 and the results are provided in Table 31. Seven projects (ESC Record Numbers 13.1, 13.4, 13.7, 14.1, 14.5, 15.1,15.2,. 34.3, and 34.6 were eliminated because their calibrated values fell outside the observed range of reasonable values identified in Table 7.

This category did not respond well to calibration, and it failed to meet the criteria of Conte, Dunsmore, and Shen for estimating accuracy. The default model is more accurate than the calibrated model (i.e., assuming the use of the mean Ctb calibration value of 5384.6) for 9 of the 17 projects, which provides insight as to why the calibration showed a decline in estimating accuracy from the default model. As discussed in the previous sections, the calibrated model's accuracy becomes worse as the range of calibration values gets larger. For this category, the individual calibration values ranged from 2173.52 (Project 13.2) to 8113.68 (Project 14.2). The composite calibration values, as indicated in Appendix D, ranged from 5060.72 (10% level) to 5719.17 (90% level). Only one project has an individual calibration value that falls within the composite value range.

Table 30. ESC Contractor J Projects Used for Calibration and Validation

SMC Record #	Size (S)	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 13.2	6060	12.00	2913.80	5001.00	1.72	30.72	83.50	2,173.52
Project 13.3	79300	12.00	1210.20	4499.60	3.72	1929.74	1154.40	6,904.39
Project 13.5	57750	15.00	2796.40	6658.60	2.38	527.84	421.00	8,039.63
Project 13.6	21750	12.00	4014.60	5859.20	1.46	96.91	325.60	2,134.27
Project 14.2	63350	12.00	2656.90	5995.50	2.26	573.65	399.00	8,113.68
Project 14.3	9078	12.00	2652.10	5995.50	2.26	55.86	107.00	3,487.96
Project 14.4	13313	15.00	2447.60	4690.20	1.92	106.46	122.00	4,186.93
Project 15.3	5400	12.00	2910.60	6274.70	2.16	26.79	54.60	3,466.11
Project 15.4	8778	15.00	2350.90	6274.70	2.67	67.79	77.80	5,594.23
Project 34.2	72345	8.00	2006.20	4499.60	2.24	801.31	1352.00	2,909.80
Project 34.4	23000	15.00	3999.00	7178.90	1.80	113.84	108.00	7,500.96
Project 34.7	31795	12.00	3246.00	6658.60	2.05	197.25	157.00	8,053.33
Project 38.2	49600	15.00	3308.60	6658.60	2.01	359.39	343.00	6,922.73
Project 38.3	18885	12.00	2791.10	5434.60	1.95	126.53	160.00	4,469.32
Project 38.4	52925	12.00	2088.30	4730.40	2.27	617.22	400.00	6,790.18
Total Projects	15							

Table 31. ESC Contractor J, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.36	0.44	22.2% worse	MMRE<0.25
RRMS	0.75	0.68	9.3% better	RRMS<0.25
Pred Interval (25%)	0.33	0.03	90.9% worse	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.37	0.47	27.0% worse	MMRE<0.25
RRMS	0.47	0.57	21.3% worse	RRMS<0.25
Pred Interval (25%)	0.33	0.14	57.6% worse	Pred (25%)>0.75
Summary information				
Number Projects Used	17			
Mean Ctb Value	5384.6			

* Represents mean value resulting from the simulation run

Contractor R (ESC)

A total of 6 projects was used for the calibration and validation of Contractor R in the ESC database. The projects used are identified in Table 32 and the results are provided in Table 33. Three projects (ESC record numbers 25.4, 26.4, and 26.5) were eliminated because their calibrated values fell outside the observed range of reasonable values identified in Table 7.

This category responded well to calibration. In addition, the calibrated model met two of the three criteria identified by Conte, Dunsmore, and Shen for estimating accuracy. The calibrated MMRE for the validation data set was 0.21 and the RRMS was 0.23, each indicating better results than the target objective of 0.25. The mean prediction interval was a reasonable 0.54, but below the target value of 0.75. This is a good example of a category that demonstrates some range in calibration values (3910.50 to 7046.86) but has a relatively tight core of calibration values for most of the projects. The mean calibration value of 5650.44 for the category serves as a good predictor, especially for projects 25.2, 26.2., 26.3 which have individual values that are tightly packed around it. It seems reasonable to suggest that the management of Contractor R explore the projects that have the extreme calibration values (projects 26.1 and 25.5) to examine what one team did well and that the other did poorly, if the calibration values indeed reflect the true project team performance.

Table 32. ESC Contractor R Projects Used for Calibration and Validation

ESC Record #	Size (S)	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 25.1	21300	9.00	2153.20	6911.90	3.21	177.90	186.80	6,636.38
Project 25.2	33400	9.00	3333.20	8102.30	2.43	180.67	279.60	5,630.73
Project 25.5	22000	9.00	2524.00	8102.30	3.21	152.84	180.70	7,046.86
Project 26.1	18124	15.00	4166.20	8729.30	2.10	81.43	213.44	3,910.50
Project 26.2	27440	15.00	4451.60	6749.40	1.52	123.71	165.20	5,303.88
Project 26.3	37183	15.00	5757.40	8729.30	1.52	130.83	237.60	5,309.18
Total Projects	6							

Table 33. ESC Contractor R, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.31	0.17	45.2% better	MMRE<0.25
RRMS	0.39	0.19	51.3% better	RRMS<0.25
Pred Interval (25%)	0.34	0.67	97.1% better	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.32	0.21	34.4% better	MMRE<0.25
RRMS	0.36	0.23	36.1% better	RRMS<0.25
Pred Interval (25%)	0.32	0.54	68.8% better	Pred (25%)>0.75
Summary information				
Number Projects Used	6			
Mean Ctb Value	5650.44			

* Represents mean value resulting from the simulation run

Other Results from Model Calibration and Validation

Appendix D provides additional results from the Monte Carlo simulation trials that were used in the calibration and validation the SAGE model. For each calibration category, the 10%, 30%, 50%, 70%, and 90% confidence levels and the mean score are provided for: 1) the composite or categorical Ctb calibration value; and 2) the Conte, Dunsmore, and Shen criteria used to measure the calibrated model accuracy for the validation data points. The 10% confidence level, for example, can be considered as meaning that 10% of the simulation runs returned the value indicated in the appendix or less. Consider, for illustration, the Mil-Spec Avionics category. In this category, 10% of the simulation runs returned a MMRE value of 0.21 or better for the validation data points.

This information augments the results provided previously in this chapter in which only mean levels were reported. Note also that the determination in previous sections of whether the SAGE model met Conte, Dunsmore, and Shen's criteria was based on the mean value of the results. Appendix D provides the additional detail to make this assessment at several confidence levels.

Summary

Three of the six research objectives proposed in Chapter I are addressed through the results provided in this chapter. These objectives were: 1) to measure the improvement, if any, from calibrated SAGE estimates as compared to uncalibrated estimates; 2) to determine whether the model meets the criteria of an effective estimating model and; 3) to examine whether calibrating the model to a specific contractor's capabilities is more appropriate than calibrating the model to a class of programs.

The first two of these objectives have already been discussed in some detail. The effects of calibration were mixed. In only three of the ten selected categories did

calibration improve model estimating accuracy as measured by the Conte, Dunsmore, and Shen criteria. Two of the ten categories became worse. The model also did not meet the proposed criteria of an effective model; however, as will be discussed further in Chapter 5, it is conceivable that the proposed criteria are excessively high for the research environment in which the SAGE model has been used and calibrated.

With respect to the third objective, there is evidence that suggests it is more appropriate to calibrate a model on an organizational basis. The SAGE model was more responsive to calibration that involved the three contractors from the ESC database than it was to calibration that involved the seven program categories from the SMC database. The primary reason why calibration by contractor was more successful centered on the tighter range of calibration values for the individual projects of a given contractor. Notionally, a tight range lends some confidence that the projects come from a comparable environment. In the SMC categories in which a large range of calibration values is present, there is doubt as to whether the wide disparity exists because of true differences in the development environmental factors that the Ctb value measures. For example, consider the calibration values of Projects 302 and 346 in the Mil-Spec Avionics category. Project 346 has a below average rating of 3482.41 while Project 302 has an above average rating of 9663.10. It may be the case that the organization that developed Project 302 had a much better development environment in place. It is also likely though that the difference in effort for Projects 302 and 346 is attributable in part to non-environmental factors that neither the data nor the model captures, but which is nonetheless "explained" by the wide-disparity in calibration values.

The above discussion is not to imply that the objective of a calibration is to tighten the gap in calibration values as much as possible. Statistically, a very tight band of values would lead to extremely good calibration results because the average calibration value for the range would serve as a reasonable estimate for all the values that are so

tightly packed around it. In reality, however, software development teams do demonstrate varying levels of performance and it is expected that these differences manifest themselves in some range of calibration values. In fact, software development managers can gain considerable insight by identifying those projects at the extremes and capitalizing on one project team's successes and avoiding another's failures. The crux of the problem, therefore, is to identify categories in which a reasonable range of calibration values exists, such that valid comparisons between software development teams can be made. The results of this research provides some indication that calibration by contractor provides a better foundation for reasonable comparisons.

V. Conclusions and Recommendations

Overview

The results of this research are consistent with the majority of results from previous efforts at the Air Force Institute of Technology to calibrate software cost estimating models. Despite moderate success in some categories, the calibrated SAGE Cost/Schedule Estimating model does not meet the criteria proposed by Conte, Dunsmore, and Shen for estimating accuracy. This does not deny, however, that the SAGE model can be extremely useful in estimating software development effort or that calibration is a productive activity. This chapter briefly reviews the findings of Chapter IV, and then proposes an analogy to put these findings in context. The analogy illustrates why the environment in which this model has been calibrated is not ideal and how this subsequently contributes to estimating error and calibration ineffectiveness. This chapter also examines how the SAGE model, regardless of its accuracy in this particular effort, nonetheless provides a foundation for understanding both the software development process and the impact that any number of inputs to the process have on the output. Finally, this chapter proposes areas for future related research.

Review of Findings

Table 34 and Table 35 that follow summarize both the default and calibrated model performance for the SMC and ESC projects used in validation. As discussed in Chapter IV, the model did not meet all proposed criteria for estimating accuracy in any of the calibration categories; however, the model performed reasonably well in two categories: Military Ground - Command and Control and ESC Contractor R. In addition, the fact that the greater proportion of the ESC database categories used benefited from calibration lends some credence to the process of calibration by contractor.

Table 34. Model Performace for SMC Validation Projects

Avionics (9 Projects)			
	Default	**Calibrated**	**Criteria Met?**
MMRE	0.45	0.39	No/No
RRMS	0.54	0.52	No/No
Pred (25%)	0.21	0.24	No/No
Mil Ground - Command & Control (10 Projects)			
MMRE	0.23	0.29	Yes/No
RRMS	0.23	0.30	Yes/No
Pred (25%)	0.70	0.45	No/No
Mil Ground - Signal Processing (16 Projects)			
MMRE	0.39	0.50	No/No
RRMS	0.43	0.54	No/No
Pred (25%)	0.44	0.20	No/No
Unmanned Space (7 Projects)			
MMRE	0.66	0.59	No/No
RRMS	0.69	0.88	No/No
Pred (25%)	0.14	0.30	No/No
Ground In Support Of Space (14 Projects)			
MMRE	0.32	0.32	No/No
RRMS	0.44	0.44	No/No
Pred (25%)	0.43	0.43	No/No
Military Mobile (10 Projects)			
MMRE	0.37	0.41	No/No
RRMS	0.47	0.52	No/No
Pred (25%)	0.29	0.36	No/No
Missile (4 Projects)			
MMRE	0.66	0.67	No/No
RRMS	0.89	0.44	No/No
Pred (25%)	0.00	0.24	No/No

Table 35. Model Performance for ESC Validation Projects

	Default	**Calibrated**	**Criteria Met?**
Contractor A (17 Projects)			
MMRE	0.48	0.41	No/No
RRMS	0.57	0.40	No/No
Pred (25%)	0.17	0.31	No/No
Contractor J (17 Projects)			
MMRE	0.37	0.47	No/No
RRMS	0.47	0.57	No/No
Pred (25%)	0.33	0.14	No/No
Contractor R (6 Projects)			
MMRE	0.32	0.21	No/Yes
RRMS	0.36	0.23	No/Yes
Pred (25%)	0.32	0.54	No/No

A theme that was discussed at some length in Chapter IV indicated that a wide range in calibration values within a category contributed to poor calibrated model performance. At issue is whether the wide range truly represents large differences in project team performance or whether the range is the result of other factors, such as incomparable project environments. The intent of stratifying the data into these categories was to gain a degree of comparability. Considering the wide range in values, however, there remains doubt whether this objective was met.

In some instances, additional steps can be taken to increase confidence that the projects come from homogenous environments. If sufficient data exist, further stratification of the categories can be accomplished. For example, in the SMC category, Ground in Support of Space, it was noticed that a subset of nine projects involved the Command/Control function. As reported in Appendix C, the calibrated model for this subset of nine projects performed extremely well, meeting each of the criteria for estimating accuracy. Nevertheless, even if several projects from comparable environments can be grouped satisfactorily, model calibration still poses a formidable challenge. Essentially, the challenge is presented by the environment in which the calibration is being performed. This challenge is now explored through the use of analogy.

Model Calibration Analogy

Overview. An analogy is proposed here to examine an ideal environment for model calibration, and then compare this ideal to the environment for this effort. The point of this analogy is to illustrate both the motivation that underlies the need to calibrate models and the sensitivity of calibration to a given environment. The analogy considers the theoretical case of a model that has been developed to predict academic performance of undergraduate students.

Theoretical Case of Calibration. The dean of the postgraduate school for a major university has identified the need to better predict the performance of students entering the university's advanced programs in government, engineering, and business. A colleague recommends a model that has been used with great success in the university's undergraduate programs. The model was developed by the university's research department and validated with data from 10 years of student records. The colleague explains that the model considers such factors as the student's prior academic grade point average (GPA), standardized test scores, academic major complexity, study habits, social habits, family history, motivational influences, and a host of other factors. The model accurately predicts student performance, as measured by the student's undergraduate GPA, within 25%. The dean is familiar with the model and finds that it captures many of the elements that he has found to be important contributors to student success during his 20 years in academia.

The dean also realizes that undergraduate and graduate academic programs are of different natures. At the surface, this difference manifests itself in the fact that graduates from postgraduate programs, on average, have higher GPAs than those from undergraduate programs. The dean understands that if he wants to use the model effectively for his advanced degree programs then he must calibrate it to his environment. The dean, calling upon his vast experience, develops a calibration methodology. He dismisses a method that involves a linear multiplier that functionally equates an undergraduate's GPA to a graduate's GPA because he believes the conversion is more complex. His belief is that graduate students, on average, are more intelligent and are more motivated than undergraduates and this translates into higher grades. As such, his methodology involves calibrating the model's inputs that account for these factors.

The dean's effort to calibrate the model is enhanced by his ability to access the registrar's records and student profiles available in the student's application package.

The dean is also quite familiar with each program's various disciplines and faculty. For example, he knows that the nuclear engineering option is considerably more challenging than the other engineering specialty options and that he must establish separate calibration factors for predicting nuclear engineering student performance. After some fine-tuning, the dean develops calibration factors for each segregated discipline that yield accuracy results comparable to the 25% error experienced by the model when used to estimate performance of undergraduate students.

Analogy Explored. The preceding example is rather simple but is illustrative in several respects. First, it provides an understandable motive of why someone would want to calibrate a model at all. The undergraduate model captures many of the same factors of student success that are also present in the graduate school environment but it does so to varying extents and with a different quantification of student performance. Calibration allows the dean to re-cast these factors to provide more reasonable outputs (i.e., student performance) for his environment. This is the same intent of the AFIT research to calibrate software cost models. Second, the environment in which the dean calibrated the model is ideal when compared to the environment in which SAGE was calibrated here. Specifically, differences can be seen in: 1) the dean's experience with the environment in general; 2) the dean's experience with the specific university environment that he is calibrating; and 3) the completeness and consistency of the data used to calibrate the model. Each difference is examined below.

First, although the dean is only somewhat familiar with the model itself, his experience in academia gives him significant insight into the interaction of student characteristics and student performance. He has an intuitive sense of the relative importance of these factors and this helps him understand the model's behavior, outputs, sensitivities, strengths, and weaknesses. On the other hand, this researcher has limited experience in software development and cost estimation. In light of this inexperience, a

detailed analysis of the behavior of SAGE's fundamental schedule and effort estimating equations was considered and presented in Chapter II; however, it is provided from purely an academic perspective. The researcher's use of the model is not enhanced by any significant experience in the field. The impact that inexperience might play on calibration results is intangible but it is assumed that practical experience can only augment model use and calibration in a positive way.

Second, the dean has exclusive insight into the environment that he is calibrating. For example, he is able to isolate nuclear engineering from the other engineering disciplines because these students exhibit a different pattern of performance. He is also more apt to identify and account for data anomalies, such as a subset of students whose performance was hindered by an ineffective faculty member or by some other extenuating circumstances. This calibration of SAGE, as well as other AFIT calibration efforts, have been performed essentially one step removed from the environment from which the data are taken. Except for some broad guidelines that are used to create calibration categories and to eliminate inappropriate data points, the data from SMC and ESC, in essence, must be accepted at face value. The potential exists that anomalies that may have been identified by a first-hand observer are not picked up and these anomalies subsequently induce erratic calibration results.

Finally, the dean benefits from having complete data on which to perform the calibration. In this SAGE effort, incomplete information for some data points forces the calibration factor to explain differences in output that may have been attributable to other factors, for which information is not available. The dean, on the other hand, can draw upon records of hundreds of students and eliminate incomplete records. The option of eliminating incomplete records is not possible for this effort, considering the small base of data to start. As discussed in Chapter IV, wide variation in calibration factors, as

caused by incomplete data records, can cause the calibrated model to return worse results than the default model.

Analogy Conclusion. The point of this analogy is not to dismiss the results of this particular effort, but rather to illustrate the fundamental need for calibration and to highlight the sensitivity of model calibration to the environment in which it is performed. Even if calibration were performed in a more ideal environment, any model will still have an estimating error that the model user must decide is acceptable for his or her estimating purposes. The estimator must be cognizant of the model's limits when using the model's output to make programmatic decisions.

Regardless of the results of this or any of the other calibration efforts, a less obvious benefit of model calibration is that it forces the user, who might otherwise treat the estimating model as a black box, to consider the underlying model behavior, to be aware of cost drivers, and thus better understand the structure of the software development process. This facet is now explored.

Other Benefits of Software Cost Model Use

Until this point, it has been presumed that the sole purpose of calibrating the SAGE model has been to obtain more accurate estimates of effort. The readers of this and related AFIT efforts might tend to dismiss many of the models solely on the basis of less than stellar results. The preceding analogy was offered to place this effort in a context that can help explain extenuating factors that burden the model's normal estimating error. A more subtle benefit of model use that only tangentially pertains to estimating accuracy for a given environment, is that cost models provide a structure for understanding key elements of the software development process.

The SAGE model, or any other of the available cost estimating models for that matter, can be considered a box in which the model developer has captured years of

experience in the field. The inputs that the model considers are those that the developer considers as essential explanatory variables for software development effort and schedule. The fact that, collectively, the models consider many of the same elements suggests that these factors must be given due consideration when performing an estimate. On the other hand, each model generally offers a unique perspective of software development. In SAGE, Dr. Jensen emphasizes effective management of development teams. SAGE model ratings are unique in that many are not defined in terms of individual characteristics, but whether the individuals work in an organization that has an effective management approach. By gaining experience with this and other models, the estimator can establish a strong framework for understanding the critical explanatory factors and gain an appreciation of the various approaches that have been proposed to better develop software. Through this "awakening", estimating accuracy should improve. This serves the estimator well, not only when using the algorithmic models, but also when using any of the other estimating techniques presented in Chapter II. For example, the estimator who decides to conduct a bottoms-up estimate might use knowledge gained from the model as a guide for the more detailed cost analysis of that particular project.

Calibration can further this level of understanding. Calibration lends insight into how the relationship of inputs to outputs changes as the model is used outside of its originally intended environment. This forces the estimator to consider the unique aspects of his or her own environment and how these particular aspects relate to development effort and schedule. In essence, calibration is a dynamic process through which not only is the model fine-tuned to an environment, but the estimator becomes better attuned as well. Although the various models are typically not marketed on the basis that they improve the user's understanding of software development, this certainly is an intrinsic value of the models that cannot be casually ignored.

Recommendations for Future Research

The focus of AFIT research to date has been on evaluating and improving the accuracy of the software estimating models in a research environment. The basis of two of the recommendations that follow is to study how these models are actually used in the field and to gauge their accuracy in more of a realistic setting. The third recommendation is to consider the further application of resampling methods proposed in this effort to generate more robust calibration results. The final suggestion is to reconsider at least a portion of the previous SMC calibrations and perform them by stratifying the projects primarily by contractor.

The first suggestion involves studying how the software cost estimating methodologies are employed in the field. In Chapter II, the various methodologies are presented and described. It is recommended by some authorities that these methods (e.g., analogy, parametric, bottoms-up) be used together to some degree to improve the estimate. There appears, however, to be no significant research that addresses how these various methods are actually employed by estimators. The proposed research would examine field practices to determine if better estimates result when two or more of the methodologies are used in combination.

The second recommendation is to measure software cost model accuracy when the model has been calibrated by an estimator who is experienced in the environment that is being calibrated, who has more complete data sets, and who has more insight into the data used to calibrate. It has been proposed in this and the previous chapter that environment inexperience as well as incomplete and perhaps anomalous data add to the model's normal estimating error.

The third and fourth recommendations may be considered in tandem. This effort proposed resampling using Monte Carlo simulation as the means to generate more robust calibration results; however, the potential exists to further capitalize on this concept.

Specifically, the optimal split of data to use for calibration and validation can be explored. In this effort, two projects were held out for validation; however, a different mix might be more suitable. In addition, effective ways to present the data that results from a simulation run can be explored. This improved use of resampling can be coupled with the re-examination of previous AFIT efforts to calibrate software cost models using data from SMC. The first step would be to persuade the maintainers of the SMC SWDB to add a database field that indicates the developing contractor. The next step would be to re-accomplish calibration for at least a portion of these models with a primary stratification by contractor. The objective would be to see if this method is more suitable than that previously employed.

Summary

This chapter has examined the use and calibration of software cost models from a philosophical perspective. The challenges of calibration are examined through the use of an analogy that helps portray the sensitivity of calibration to the environment in which it is performed. Other benefits of model use and calibration are explored, specifically with respect to the foundation models can provide for understanding software development processes and perspectives. Finally, in response to the sixth and final research objective proposed in Chapter I, future efforts in this research area are proposed.

Appendix A: SMC SWDB and SAGE Input Value Correlation Matrix

	Very Low	Low	Nom.	High	Very High	Extra High
SMC Database Title (Column Number)						
SAGE Rating						
Inherent Difficulty of Application (4.23.1)						
Complexity (D)	28	21	15	12	8	4
Personnel Capability (4.8.2)						
ACAP	1.46	1.19	1	0.86	0.71	
Personnel Experience (4.8.1)						
AEXP (D>13)	1.29	1.13	1	0.91	0.82	
AEXP (10<D<13)	1.29	1.13	1	0.91	0.82	
AEXP (D<10)	1.29	1.13	1	0.91	0.82	
Host Virtual System (4.8.4)						
DEXP Code		1	2	3		
Development System Experience (4.8.7)						
DEXP-1	1.11	1.04	1	1	1	1
DEXP-2	1.16	1.1	1.04	1	1	1
DEXP-3	1.21	1.16	1.08	1.03	1	1
Team Programming Language Experience (4.8.5)						
LEXP 0.5	1.15	1.02	1	1	1	1
LEXP 1.0	1.2	1.07	1	1	1	1
LEXP 2.0	1.28	1.17	1.06	1	1	1
LEXP 3.0	1.37	1.26	1.13	1.05	1	1
LEXP 5.0	1.53	1.4	1.2	1.09	1.04	1
Personnel Capability (4.8.2)						
PCAP	1.42	1.17	1	0.86	0.7	
Development Methods Experience (4.8.6)						
PEXP	1.29	1.11	0.93	0.85	0.83	0.83
Target Virtual System (4.8.3)						
TEXP Code		1	2	3		
Target System Experience (4.8.8)						
TEXP-1	1.06	1.02	1	1	1	
TEXP-2	1.09	1.06	1.02	1	1	
TEXP-3	1.11	1.09	1.05	1.02	1	
Modern Practices Experience (4.23.13)						
MODP	1.21	1.1	1	0.91	0.83	
Automated Tool Support (4.23.14)						
TOOL	1.24	1.1	1	0.91	0.83	
Resource/Support Location (4.23.10)						
RLOC			1	1.12	1.23	1.35
Reuse Impact (N/A)						
RIMP			1			

	Very Low	Low	Nom.	High	Very High	Extra High
Reusability Requirements (4.3.6)						
RUSE			1	1	1	1
Development System Volatility (4.23.4)						
DVOL		1	1.07	1.13	1.19	1.25
Resource Dedication (4.23.9)						
RDED	1.25	1.11	1			
Terminal Responses (4.23.3)						
RESP		0.96	1	1.05	1.1	1.14
Turnaround Time (4.3.2)						
TURN	0.87	0.93	1	1.07	1.15	1.15
Multiple Site Development (4.23.8)						
MULT			1	1.07	1.13	1.2
Display Requirements (4.3.5)						
DISP			1	1.05	1.11	1.16
Rehosting Requirements (4.3.4)						
HOST			1	1.24	1.63	
Memory Constraints (4.3.8)						
MEMC			1	1.04	1.15	1.4
Quality Assurance Level (4.23.6)						
QUAL	1	1.02	1.03	1.06	1.09	
Requirements Volatility (4.3.3)						
RVOL		0.93	1	1.15	1.29	1.46
Real Time (4.3.10)						
RTIM			1	1.04	1.09	1.27
Security Level (4.3.7)						
SECR			1	1.4	2.1	3.82
Specification Level (4.23.5)						
SPEC		1	1.12	1.21	1.26	1.28
Test Level (4.23.7)						
TEST	1	1.02	1.05	1.11	1.27	
Timing Constraints (4.3.9)						
TIMC			1	1.1	1.26	1.56

Appendix B: ESC Database and SAGE Input Value Correlation Matrix

	Very Low	Low	Nom.	High	Very High	Extra High
ESC Database Title (Abbreviation)						
SAGE Rating						
Complexity (CPLX)						
Complexity (D)	28	21	15	12	8	4
Analyst Capability (ANAL CAP)						
ACAP	1.46	1.19	1	0.86	0.71	
Analyst Experience (ANAL EXP)						
AEXP (D>13)	1.29	1.13	1	0.91	0.82	
AEXP (10<D<13)	1.29	1.13	1	0.91	0.82	
AEXP (D<10)	1.29	1.13	1	0.91	0.82	
Host Virtual System (HOST CPLX)						
DEXP Code		1	2	3		
Development System Experience (HOST EXP)						
DEXP-1	1.11	1.04	1	1	1	1
DEXP-2	1.16	1.1	1.04	1	1	1
DEXP-3	1.21	1.16	1.08	1.03	1	1
Team Programming Language Experience (LANG EXP)						
LEXP 0.5	1.15	1.02	1	1	1	1
LEXP 1.0	1.2	1.07	1	1	1	1
LEXP 2.0	1.28	1.17	1.06	1	1	1
LEXP 3.0	1.37	1.26	1.13	1.05	1	1
LEXP 5.0	1.53	1.4	1.2	1.09	1.04	1
Programmer Capability (PROG CAPB)						
PCAP	1.42	1.17	1	0.86	0.7	
Development Methods Experience (MODP EXP)						
PEXP	1.29	1.11	0.93	0.85	0.83	0.83
Target Virtual System (TRGT CPLX)						
TEXP Code		1	2	3		
Target System Experience (TRGT SEXP)						
TEXP-1	1.06	1.02	1	1	1	
TEXP-2	1.09	1.06	1.02	1	1	
TEXP-3	1.11	1.09	1.05	1.02	1	
Modern Practices Experience (MODP EXP)						
MODP	1.21	1.1	1	0.91	0.83	
Automated Tool Support (TOOL USE)						
TOOL	1.24	1.1	1	0.91	0.83	
Practices Volatility (MODP VOL)						
PVOL						
Resource/Support Location (RES LOC)						
RLOC			1	1.12	1.23	1.35

	Very Low	Low	Nom.	High	Very High	Extra High
Reuse Impact (N/A)						
RIMP			1			
Reusability Requirements (N/A)						
RUSE			1	1	1	1
Development System Volatility (HOST VOL)						
DVOL		1	1.07	1.13	1.19	1.25
Resource Dedication (RES DED)						
RDED	1.25	1.11	1			
Terminal Responses (TERM RESP)						
RESP		0.96	1	1.05	1.1	1.14
Turnaround Time (TURN TIME)						
TURN	0.87	0.93	1	1.07	1.15	1.15
Multiple Classifications (N/A)						
MCLS			1			
Multiple Organizations (N/A)						
MORG			1			
Multiple Site Development (N/A)						
MULT			1			
Display Requirements (SPEC DISP)						
DISP			1	1.05	1.11	1.16
Rehosting Requirements (REHOST)						
HOST			1	1.24	1.63	
Memory Constraints (MEM CONST)						
MEMC			1	1.04	1.15	1.4
Quality Assurance Level (QA LEVL)						
QUAL	1	1.02	1.03	1.06	1.09	
Requirements Volatility (N/A)						
RVOL			1			
Real Time (Real Time)						
RTIM			1	1.04	1.09	1.27
Security Level (SEC REQ)						
SECR			1	1.4	2.1	3.82
Specification Level (SPEC LVL)						
SPEC		1	1.12	1.21	1.26	1.28
Test Level (TST LVL)						
TEST	1	1.02	1.05	1.11	1.27	
Timing Constraints (TIME CONST)						
TIMC			1	1.1	1.26	1.56

Appendix C: Ground in Support of Space - Command/Control Calibration

Ground in Support of Space - Comm/Control Projects Used for Calibration

SMC Record #	Size	Cmplx (D)	Cte (SAGE Estimate)	Ctb (SAGE Estimate)	Ctb/Cte	Model Estimate of Effort	Actual Effort	Calibrated Ctb
Project 74	11700	12.00	2303.70	5158.50	2.24	89.68	80.00	5,673.8
Project 75	116800	8.00	1785.80	5158.50	2.89	1637.17	912.00	8,399.8
Project 76	14000	15.00	2166.60	5158.50	2.38	130.91	115.00	5,746.8
Project 77	56200	8.00	1785.80	5158.50	2.89	680.53	523.00	6,424.0
Project 78	48300	15.00	2303.70	5158.50	2.24	537.51	478.00	5,688.4
Project 79	50300	15.00	2303.70	5158.50	2.24	564.33	432.00	6,445.1
Project 81	22900	15.00	2303.70	5158.50	2.24	219.51	164.00	6,577.0
Project 82	16300	15.00	2303.70	5158.50	2.24	145.97	140.00	5,341.3
Project 83	6800	21.00	2303.70	5158.50	2.24	58.49	57.00	5,270.9
Total Prj	9							

Ground in Support of Space - Command/Control, Model Accuracy Results

For data used to calibrate the model				
Measurement	**Default Model***	**Cali. Model***	**Change from Calibration**	**Target**
MMRE	0.24	0.13	45.8% better	MMRE<0.25
RRMS	0.72	0.37	48.6% better	RRMS<0.25
Pred Interval (25%)	0.56	0.89	58.9% better	Pred (25%)>0.75
For data used to validate the model				
MMRE	0.25	0.16	36.0% better	MMRE<0.25
RRMS	0.39	0.21	46.2% better	RRMS<0.25
Pred Interval (25%)	0.55	0.89	61.8% better	Pred (25%)>0.75
Summary information				
Number Projects Used	9			
Mean Ctb Value	6172.58			

Appendix D: Other Simulation Results for SMC and ESC Calibration

	Confidence Level						
Mil-Spec Avionics	**10%**	**30%**	**50%**	**Mean**	**70%**	**90%**	**Target**
Composite Ctb Value	6018.44	6330.66	6525.60	6545.32	6744.06	7154.01	
MMRE*	0.21	0.30	0.40	0.39	0.45	0.57	MMRE < 0.25
RRMS*	0.27	0.42	0.53	0.52	0.57	0.80	RRMS < 0.25
Pred Interval (25%)*	0.00	0.00	0.00	0.24	0.50	0.50	Pred > 0.75
# of Projects	9						
Simulation Trials	3600						
Mil Grd - Comm & Control	**10%**	**30%**	**50%**	**Mean**	**70%**	**90%**	**Target**
Composite Ctb Value	5568.59	5732.62	5902.71	5912.02	6005.70	6239.79	
MMRE*	0.17	0.22	0.26	0.29	0.36	0.46	MMRE < 0.25
RRMS*	0.14	0.23	0.28	0.30	0.36	0.46	RRMS < 0.25
Pred Interval (25%)*	0.00	0.50	0.50	0.45	0.50	1.00	Pred > 0.75
# of Projects	10						
Simulation Trials	7200						
Mil Grd - Signal Processing	**10%**	**30%**	**50%**	**Mean**	**70%**	**90%**	**Target**
Composite Ctb Value	4058.79	4209.43	4306.06	4301.68	4404.87	4581.60	
MMRE*	0.27	0.39	0.48	0.50	0.60	0.79	MMRE < 0.25
RRMS*	0.30	0.38	0.53	0.54	0.63	0.85	RRMS < 0.25
Pred Interval (25%)*	0.00	0.00	0.00	0.20	0.50	0.50	Pred > 0.75
# of Projects	16						
Simulation Trials	10,000						
Ground in Support of Space	**10%**	**30%**	**50%**	**Mean**	**70%**	**90%**	**Target**
Composite Ctb Value	4929.02	5065.14	5156.16	5176.07	5292.31	5383.34	N/A
MMRE*	0.09	0.22	0.27	0.32	0.38	0.58	MMRE < 0.25
RRMS*	0.11	0.31	0.40	0.44	0.52	0.88	RRMS < 0.25
Pred Interval (25%)*	0.00	0.00	0.50	0.43	0.50	1.00	Pred > 0.75
# of Projects	14						
Simulation Trials	9,100						
Military Mobile	**10%**	**30%**	**50%**	**Mean**	**70%**	**90%**	**Target**
Composite Ctb Value	4703.57	4902.21	5052.73	5060.08	5258.32	5435.08	N/A
MMRE*	0.16	0.31	0.40	0.41	0.50	0.62	MMRE < 0.25
RRMS*	0.16	0.27	0.37	0.52	0.69	0.99	RRMS < 0.25
Pred Interval (25%)*	0.00	0.00	0.50	0.36	0.50	1.00	Pred > 0.75
# of Projects	10						
Simulation Trials	4,500						

*These are the results for the calibrated model accuracy for the validation data points.

Missiles	10%	30%	50%	Mean	70%	90%	Target
Composite Ctb Value	1737.74	1764.16	2877.79	2366.29	2948.17	2974.59	N/A
MMRE*	0.42	0.46	0.48	0.67	0.89	0.94	MMRE < 0.25
RRMS*	0.19	0.39	0.46	0.44	0.52	0.60	RRMS < 0.25
Pred Interval (25%)*	0.00	0.00	0.00	0.24	0.50	0.50	Pred > 0.75
# of Projects	4						
Simulation Trials	600						
Unmanned Space	**10%**	**30%**	**50%**	**Mean**	**70%**	**90%**	**Target**
Composite Ctb Value	1680.73	1912.09	2080.55	2013.59	2264.84	2281.71	N/A
MMRE*	0.24	0.29	0.30	0.59	0.42	1.17	MMRE < 0.25
RRMS*	0.28	0.29	0.30	0.88	0.46	2.11	RRMS < 0.25
Pred Interval (25%)*	0.00	0.00	0.50	0.30	0.50	0.50	Pred > 0.75
# of Projects	7						
Simulation Trials	2,100						
ESC Contractor A	**10%**	**30%**	**50%**	**Mean**	**70%**	**90%**	**Target**
Composite Ctb Value	3707.72	3836.11	3919.88	3901.00	3986.76	4077.94	N/A
MMRE*	0.16	0.28	0.37	0.41	0.47	0.79	MMRE < 0.25
RRMS*	0.18	0.31	0.39	0.40	0.49	0.61	RRMS < 0.25
Pred Interval (25%)*	0.00	0.00	0.5	0.31	0.50	0.50	Pred > 0.75
# of Projects	17						
Simulation Trials	10,000						
ESC Contractor J	**10%**	**30%**	**50%**	**Mean**	**70%**	**90%**	**Target**
Composite Ctb Value	5060.72	5265.11	5410.48	5384.60	5511.60	5719.17	N/A
MMRE*	0.28	0.40	0.48	0.47	0.57	0.64	MMRE < 0.25
RRMS*	0.36	0.46	0.56	0.57	0.69	0.79	RRMS < 0.25
Pred Interval (25%)*	0.00	0.00	0.00	0.14	0.50	0.50	Pred > 0.75
# of Projects	17						
Simulation Trials	10,000						
ESC Contractor R	**10%**	**30%**	**50%**	**Mean**	**70%**	**90%**	**Target**
Composite Ctb Value	5289.98	5392.60	5720.04	5650.44	5806.12	6154.46	N/A
MMRE*	0.05	0.15	0.20	0.21	0.28	0.33	MMRE < 0.25
RRMS*	0.06	0.17	0.24	0.23	0.31	0.35	RRMS < 0.25
Pred Interval (25%)*	0.00	0.50	0.50	0.54	0.50	1.00	Pred > 0.75
# of Projects	6						
Simulation Trials	1,500						

*These are the results for the calibrated model accuracy for the validation data points.

References

Andolfi, M. et al. "A Multicriteria-based Methodology for the Evaluation of Software Cost Estimation Models and Tools," CSELT Technical Reports, Volume XXIV: 643-659 (August 1996).

Apgar, Henry and others. Application Oriented Software Data Collection: Software Model Calibration Report (TR-9007/49-1). Oxnard CA: Management Consulting and Research, 1991.

Conte, S. D., H. E. Dunsmore, and V. Y. Shen. Software Engineering Metrics and Models. Menlo Park CA: The Benjamin/Cummings Publishing Company, Inc., 1986.

Department of the Air Force. Guidelines for Successful Acquisition and Management of Software-Intensive Systems: Weapon Systems, Command and Control Systems, Management Information Systems. Volume 1. Hill AFB UT: Ogden Air Logistics Center, June 1996.

Ferens, Daniel V. Class handout, COST 677, Quantitative Management of Software, School of Logistics and Acquisition Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, Winter Quarter 1997.

----. "Software Cost Estimation in the DoD Environment," American Programmer, 9: 28-34 (July 1996).

Galonsky, James C. Calibration of the PRICE S Software Cost Model. MS Thesis, AFIT/GCA/LAS/95S-1. School of Logistics and Acquisition Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1995 (AD-A301337).

Gibbs, W. Wyatt. "Software's Chronic Crisis," Scientific American: 86-95, September 1994.

Humphrey, Watts S. Managing the Software Process. Reading MA: Addison-Wesley Publishing Company, 1990.

IIT Research Institute. Test Case Study: Estimating the Cost of Ada Software Development. Lanham MD: 1989.

Jensen, Randall W. "A New Perspective in Software Schedule and Estimation," Article published by Software Engineering, Inc., Brigham City UT. 1996a.

----. President, Software Engineering, Inc., Brigham City UT. Telephone Interview. 20 June 1997a.

----. President, Software Engineering, Inc., Brigham City UT. Telephone Interview. 28 April 1997b.

----. President, Software Engineering, Inc., Brigham City UT. Personal Interview. 14 February 1997c.

----. Workshop Notebook, SAGE Software Cost Estimating Workshop. Los Angeles AFB CA, November 1996b.

Jensen, Randall W. and Charles C. Tonies. Software Engineering. Englewood Cliffs NJ: Prentice-Hall, Inc., 1979.

Jones, Capers. "How Software Estimation Tools Work," American Programmer, 9: 18-27 (July 1996).

Kemerer, Chris F. "An Empirical Validation of Software Cost Estimation Models," Communications of the ACM, 30: 416-429 (May 1987).

Kressin, Robert K. Calibration of SLIM to the Air Force Space and Missile Systems Center Software Database. MS Thesis, AFIT/GCA/LAS/95S-6. School of Logistics and Acquisition Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1995 (AD-A301603).

Lucas, Suzanne. "Software Cost Estimation Using SEER-SEM Knowledge Base Capabilities," Hawthorne CA: Hughes Aircraft Company Electro-Optical and Data Systems Group. 1991.

Mertes, Karen R. Calibration of the CHECKPOINT Model to the Space and Missile Systems Center (SMC) Software Database. MS Thesis, AFIT/GCA/LAS/96S-11. School of Logistics and Acquisition Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1996 (AD-A319518).

Novak-Ley, Gina and Sherry Stukes. SMC SWDB User's Manual: Version 2.1. Oxnard CA: Management Consulting & Research, Inc., 1995.

Ourada, Gerald L. Software Cost Estimating Models: A Calibration, Validation and Comparison. MS Thesis, AFIT/GSS/LSY/91D-11. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1991 (AD-A246677)

Rathmann, Kolin D. Calibration and Evaluation of SEER-SEM for the Air Force Space and Missile System Center. MS Thesis, AFIT/GCA/LAS/95S-9. School of Logistics and Acquisition Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1995 (AD-A300703).

Shadle, Daryl A. An Investigation of the Long-term Impact of the Calibration of Software Estimation Models Using Raw Historical Data. MS Thesis. Naval Postgraduate School, Monterey CA, September 1994.

Southwell, Steven V. Calibration of the SOFTCOST-R Software Cost Model to the Space and Missile Systems Center (SMC) Software Database (SWDB). MS Thesis, AFIT/GSM/LAS/96S-6. School of Logistics and Acquisition Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1996 (AD-A319050).

Space Systems Cost Analysis Group (SSCAG). Software Methodology Handbook, Version 1.0. June 1995.

Stukes, Sherry and David Patterson. "Space and Missile Systems Center Software Database." Briefing to Thesis Students. Air Force Institute of Technology, Wright-Patterson AFB OH, 12 November 1996.

Stutzke, Richard D. "Software Estimating Technology: A Survey," Crosstalk: 17-21 (May 1996).

Thibodeau, Robert. An Evaluation of Software Cost Estimating Models. New York: Rome Air Development Center, 1981. (Contract F30602-79-C-0244)

University of Maryland. The Resampling Project. College Park MD, 1997.

Van Genuchten, Michiel and Hans Koolen. "On the Use of Software Cost Models," Information and Management 21: 37-44 (1991).

Vegas, C.D. Calibration of Software Architecture Sizing and Estimating Tool. MS Thesis, AFIT/GCA/LAS/95S-11. School of Logistics and Acquisition Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1995 (AD-A301376).

Weber, Betty G. A Calibration of the REVIC Software Cost Estimating Model. MS Thesis, AFIT/GCA/LAS/95S-13. School of Logistics and Acquisition Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1995 (AD-A300694).

Wellman, Frank. Software Costing: An Objective Approach to Estimating and Controlling the Cost of Computer Software. New York: Prentice-Hall, Inc., 1992.

Wells, Peggy. Software Database Manager, Electronic Systems Center, Air Force Material Command, Hanscom AFB MA. Telephone Interview. 29 May 97.

Vita

Captain David B. Marzo [redacted]

graduated from Spartanburg High School in 1984. In May 1988, he graduated from Georgetown University with a Bachelor of Science in Mathematics and also received his Air Force commission through the Reserve Officer Training Corps program.

Captain Marzo's first Air Force assignment was at the Human Systems Program Office (SPO), Brooks AFB, Texas from January 1989 to August 1993. While there, he served as both a project manager and the SPO's executive officer. Captain Marzo was then selected for the AFIT Education with Industry program in the project management option, where he spent ten months working for Westinghouse in Baltimore, Maryland. In July 1994, Captain Marzo moved to Wright-Patterson Air Force Base as a project manager in the Flight Training SPO, working primarily on the Joint Primary Aircraft Training System program.

Captain Marzo entered AFIT's Graduate Cost Analysis program in May 1996. Upon graduation, he anticipates a follow-on assignment to the Air Force Cost Analysis Agency in Arlington, Virginia.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1997	Master's Thesis

4. TITLE AND SUBTITLE
CALIBRATION AND VALIDATION OF THE SAGE SOFTWARE COST/SCHEDULE ESTIMATING SYSTEM TO UNITED STATES AIR FORCE DATABASES

5. FUNDING NUMBERS

6. AUTHOR(S)
David B. Marzo, Captain,USAF

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
2950 P Street
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCA/LAS/97S-6

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
SMC/FMC
2430 East El Segundo Boulevard, Suite 2010
Los Angeles, CA 90245-4687

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 Words*)

This research entailed calibration and validation of the SAGE Software Cost/Schedule Estimating System, Version 1.7 as a means to improve estimating accuracy for DoD software-intensive systems, and thereby introduce stability into software system development. SAGE calibration consisted of using historical data from completed projects at the Space and Missile Systems Center (SMC) and the Electronic Systems Center (ESC) to derive average performance factors (i.e., calibration factors) for pre-defined categories of projects. A project was categorized for calibration by either its primary application or by the contractor that developed it. The intent was to determine the more appropriate categorization for calibration. SAGE validation consisted of using the derived calibration factors to predict completed efforts, not used in deriving the factors. Statistical resampling employing Monte Carlo simulation was used to calibrate and validate the model on each possible combination of a category's projects. Three statistical measures were employed to measure model performance in default and calibrated estimating modes. SAGE generally did not meet pre-established criteria for estimating accuracy, although the model demonstrated some improvement with calibration. Calibration of projects categorized by contractor resulted in better calibrated model performance than calibration of projects categorized by application. This categorization is suggested for future consideration.

14. Subject Terms
Calibration, Cost Analysis, Cost Estimates, Cost Models, Validation, Software, Models, Cost Models, Cost Overruns, Monte Carlo Method, Sampling, Software Engineering

15. NUMBER OF PAGES
106

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

AFIT Control Number AFIT/GCA/LAS/97S-6

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. **Please return completed questionnaire** to: AIR FORCE INSTITUTE OF TECHNOLOGY/LAC, 2950 P STREET, WRIGHT-PATTERSON AFB OH 45433-7765. Your response is **important.** Thank you.

1. Did this research contribute to a current research project? a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?
a. Yes b. No

3. **Please estimate** what this research would have cost in terms of manpower and dollars if it had been accomplished under contract or if it had been done in-house.

Man Years_____ $_____

4. Whether or not you were able to establish an equivalent value for this research (in Question 3), what is your estimate of its significance?

a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments (Please feel free to use a separate sheet for more detailed answers and include it with this form):

_____ _____
Name and Grade Organization

_____ _____
Position or Title Address