A CALIBRATION OF THE REVIC SOFTWARE COST ESTIMATING MODEL

THESIS

Betty G. Weber, B.S., M.S.
Civilian, U.S. Army
AFIT/GCA/LAS/95S-13

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCA/LAS/95S-13

A CALIBRATION OF
THE REVIC SOFTWARE
COST ESTIMATING MODEL

THESIS

Betty G. Weber, B.S., M.S.
Civilian, U.S. Army
AFIT/GCA/LAS/95S-13

AFIT/GCA/LAS/95S-13

A CALIBRATION OF THE REVIC SOFTWARE COST ESTIMATING MODEL

THESIS

Presented to the Faculty of the Graduate School of Logistics

and Acquisition Management of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Cost Analysis

Betty G. Weber, B.S., M.S.

Civilian, US Army

September 1995

Approved for public release; distribution unlimited

Acknowledgments

This research could not have been completed without the aid of several individuals whose assistance I would like to acknowledge.

First, I wish to thank Professor Daniel V. Ferens, my thesis advisor, for his assistance and expertise in understanding the REVIC software model and its calibration requirements. His readily available feedback helped to smooth the many roadblocks encountered in this effort.

I would also like to thank Sherry Stukes at MCR and Gina Novak-Ley at SMC/FMC for the database which made this research possible and for their suggestions for, and assistance in, analyzing the database. Many pitfalls were avoided because of their foresight.

I would also like to thank Drs. Richard Murphy and David Christensen for their ideas, critiques, and suggestions which helped to improve the quality of my analysis.

Finally, I want to thank my husband for his computer, which made the mechanics of this research much simpler, and for his patience and support over the last sixteen months. Thanks, All.

Betty G. Weber

Table of Contents

Page

Acknowledgments ii

List of Figures v

List of Tables vi

Abstract vii

I. Introduction 1-1

Chapter Overview 1-1

General Issue 1-2

Specific Problem 1-3

Scope of Research 1-4

Summary 1-5

II. Literature Review 2-1

Chapter Overview 2-1

Summary of Cost Models 2-1

Summary of Prior Research 2-4

Summary 2-8

III. Methodology 3-1

Chapter Overview 3-1

Software Database 3-1

Cost Model 3-3

Method of Stratification 3-5

Method of Analysis 3-6

Method of Calibration 3-9

Summary 3-12

IV. Analysis and Findings 4-1

Chapter Overview 4-1

Page

Military Ground 4-1

Unmanned Space 4-7

Summary........ 4-14

V. Conclusions and Recommendations 5-1

Chapter Overview 5-1

Conclusions 5-1

Recommendations 5-5

Summary........ 5-5

Appendix A: Glossary A-1

Appendix B: Acronyms B-1

Appendix C: Military Ground Worksheets C-1

Appendix D: Unmanned Space Worksheets D-1

Bibliography........ REF-1

Vita........ V-1

List of Figures

Figure Page

4-1. Military Ground Scatter Plot 4-4

4-2. Military Ground Residuals 4-5

4-3. Military Ground NormalityTest........ 4-7

4-4. Unmanned Space Scatter Plot 4-10

4.5. Unmanned Space Residuals........ 4-11

4-6. Unmanned Space Normality Test........ 4-13

List of Tables

Table | Page

2-1. COCOMO Development Effort Algorithms 2-3

2-2. REVIC Nominal Intermediate Equations 2-4

3-1. A Comparison of Development Phase Terminology 3-4

3-2. Key to REVIC Parameters 3-5

3-3. Calibrating the Constant Term 3-10

4-1. Military Ground Calibration 4-1

4-2. Calibration Effects on Military Ground MRE 4-2

4-3. Calibration Results on Military Ground Estimates 4-3

4-4. Military Ground Wilcoxon Signed Rank Tests 4-6

4-5. Unmanned Space Calibration 4-8

4-6. Calibration Effects on Unmanned Space MRE 4-8

4-7. Calibration Results on Unmanned Space Estimates 4-9

4-8. Unmanned Space Wilcoxon Signed Rank Tests 4-12

AFIT/GCA/LAS/95S-13

Abstract

This study sought to improve cost estimating through calibration of the REVIC software cost estimating model using a database which is more completely documented than has been available heretofore. Standard regression analysis techniques, using two different methods, were used and the results of the two methodologies compared. One approach used the standard methodology described by Boehm in his book, *Software Engineering Economics*. A second approach used a standard statistics software package and a single independent variable (KDSI), ignoring the effort adjustment factor. A comparison of the results was examined.

Two separate environments were calibrated to the REVIC model, using the updated (December, 1994) Air Force Space and Missile Systems Center (SMC) software database (SWDB) containing over 2,500 records. One calibration was on a data set confined to the Military Ground operating environment and the other calibration was to the Unmanned Space operating environment. Data sets were carefully screened for completeness of information and normalized as to manhours per manmonths and to software phases included in the development. However, neither calibration produced significantly improved estimates.

The best results were obtained on the Unmanned Space database using the SAS® System for Elementary Statistical Analysis with one independent variable (KDSI) and a log-log distribution to model a linear relationship between effort and KDSI, ignoring the 19 parameters which REVIC uses as a constant multiplier. In general, when Boehm's procedure was used, better results were obtained using the simultaneous coefficient and exponent calibration then were obtained by the coefficient only calibration.

A CALIBRATION OF THE REVIC SOFTWARE COST ESTIMATING MODEL

I. Introduction

Chapter Overview

Software technology enjoys a unique position in our society's history of technology. No other technology has had such a major impact on society, businesses, research, or lifestyles in the United States. During the past thirty years, software technology has experienced a gain of six orders of magnitude in performance while simultaneously experiencing a decline in price. In no other technology can one identify such marked strides in innovation and cost (Brooks, 1987). As a result, computers and their attendant software continue to grow in popularity with customers who have come to expect better and faster performance from their software programs. When increasing popularity is combined with rapid advances in performance, the result is a scenario in which software development must struggle to keep abreast of demand.

Four years ago, the U.S. Department of Defense (DoD) established the Corporate Information Management (CIM) initiative. The CIM had, as one of its goals, the implementation of a standards-based architecture (SBA) for its information systems. The intent behind SBA was to reduce software development and support costs through reusable code, standard platforms and shared data repositories (Bozman, 1993). This initiative was followed, a year and a half later, by the establishment of the Federal High Performance Computing and Communications (HPCC) Program which has, as its stated goal, a trillion operations per second (teraops) (Nordwall, 1994). Obviously, given these combinations, the expected expansion and demand for computer hardware and software can only be expected to spiral upward.

DoD is one customer who has been quick to recognize the advantages computer technology provides to organizations and the importance of remaining on the leading edge of technological advances. As a result, DoD has been placed in the onerous position of maximizing performance while simultaneously reducing costs. With its software costs accounting for more than one-half of its $9.5 billion annual information systems budget, excluding classified systems (Bozman, 1993), DoD is spending approximately 10 percent of its total budget on software life-cycle costs (Shebalin, 1994). As the military continues to become increasingly dependent upon technology for its clout, software costs, as a percentage of the DoD budget, are expected to continue to increase at about 12 percent a year (Shebalin, 1994).

In recognition of the increasingly important role that software technology plays in U.S. national security, this thesis will address the estimation of costs for software acquisition with an emphasis on improving the accuracy of cost estimation for software development. This chapter justifies the analysis by presenting the general issues surrounding software cost models such as the Constructive Cost Model (COCOMO) and the Revised Enhanced Version of Intermediate COCOMO (REVIC) model, identifying some of the more pressing issues and the resulting research objectives which will be addressed in an effort to contribute to the resolution of the issues. This will be followed by a summary of the methodology employed with a description of its scope and limitations.

General Issue

One of the most controversial areas in software development is in the estimating of software development costs. The software development process is highly susceptible to many problems with the controversy revolving around schedule slips, cost overruns, and programs of poor quality. In 1981, a software cost estimation model was considered to be estimating well if it estimated software development costs within 20 percent of the actual costs 70 percent of the time (Boehm, 1981). Yet, despite the increasing allocation for computer research and development over the past twenty years, the accuracy of cost estimates for software development and the 20/70 success rate remains unimproved (Ourada, 1991). A large part of the problem is the trial-and-error process inherent in software development. Developers have many options in software development and may try several before they find the one that works best. As a result, software development is more unpredictable than the development of traditional hardware systems (Asbrand, 1993).

Some experts predict that future software systems will be assembled from discrete modules or components. If that occurs, software will more closely resemble hardware that is assembled from off-the-shelf parts. This would enable estimators to predict software costs with a reliability comparable to current cost estimates for hardware systems (Babcock, 1994). However, until that occurs, the need to develop better and cheaper software will continue to exist. The requirement that we improve upon our current ability to estimate software development costs is becoming more and more essential as program managers fight harder and smarter for the decreasing dollars available to support their programs.

Specific Problem

Many software cost estimating models have been developed for the purpose of estimating software development and support costs. Some of the more popular models used throughout the Air Force and DoD include the REVIC, PRICE-S, SASET, SLIM, SEER-SEM, AND SOFTCOST models.

The Space and Missile Systems Center (SMC) and DoD have a requirement for more accurate estimates for software development costs. This research proposes to examine the REVIC cost model for purposes of calibrating it to homogenous data sets. The REVIC model has been selected primarily because it is one of the more common cost models in widespread usage throughout DoD. Other factors influencing selection of REVIC for further study include REVIC's non-proprietary nature, the visibility of its algorithms, and its similarity to COCOMO and several other cost models in current use by DoD.

Scope of Research

Previous research efforts to calibrate REVIC have been constrained by limited and outdated databases. This effort will be based on a larger and more recent database, compiled, under contract, by Management Consulting & Research (MCR), Inc., and made available through the Space and Missile Systems Center (SMC) in December 1994. The research will primarily seek to determine:

1. The input parameters which most strongly influence software costs.
2. The effect of the software development environment on model performance.
3. The circumstances under which REVIC may be the most appropriate model.
4. The calibration methodology that produces the best results.
5. The extent to which calibration influences the accuracy of software estimates.

Previous efforts to calibrate REVIC have been confined to the method recommended by Boehm (Boehm, 1981). In this effort, the model will be calibrated using two methods. First the model will be calibrated using Boehm's methodology which accommodates environmental factors as constant cost multipliers (Table 3-2.). The model will be calibrated a second time using a statistical package such as SAS and assuming a log-log distribution with lines of code being the single cost driver. A comparison of the methods will be conducted to determine the influence of environmental factors on cost.

Summary

This chapter has provided an overview of the status of software development and has identified pertinent questions regarding the REVIC software cost model which this research effort will address. If it can be demonstrated that the calibration of REVIC will produce more accurate cost estimates, program managers of DoD systems will be able to more accurately predict the development costs for their systems. Not only will this result in more realistic budget requests, but will also reduce the likelihood of program cancellations due to cost overruns.

II. Literature Review

Chapter Overview

Before pursuing the research objective of calibrating REVIC, a review of the history of software development and earlier research efforts to develop or calibrate software development cost models should be considered.

Summary of Cost Models

The first significant contribution to software cost modeling was the 1965 Systems Development Corporation's (SDC's) "Nelson" model. This model was based on an analysis of 104 attributes of 169 software projects. The most conclusive result from the SDC effort was that a linear cost-estimation model would not produce useful results. Although it was not a very accurate predictor of software cost, it did produce some valuable insight into software cost estimation and served as a springboard for later models (Boehm, 1981).

One of the earliest models to enjoy a modicum of success was the TRW "Wolverton" model. It was calibrated to a class of near-real-time government command and control projects, but was less accurate for other classes of projects (Boehm, 1984). The Doty model was another early parametric model that was developed about the same time. The Doty model had some problems with stability and exhibited a discontinuity when delivered source instructions (DSI) exceeded 1,0000,000, that is when KDSI = 10,000, producing widely varying estimates (Boehm, 1984).

In the late 1970's, a major advance was made with the near simultaneous development of several software cost estimation models. Among these were the Putnam SLIM model and the RCA PRICE-S model, followed in 1981 by the COCOMO model.

The Putnam SLIM model was based on Putnam's analysis of the software life cycle in terms of the Rayleigh-Norden distribution of project personnel level versus time. The SLIM approach provided a number of useful insights into software cost estimation such as the Rayleigh curve distribution and the explicit treatment of estimation risk and uncertainty (Boehm, 1984).

RCA's (now Lockheed-Martin) PRICE-S model was a macro cost-estimation model developed primarily for embedded system applications. Early versions contained a widely varying subjective complexity factor and were primarily developed to handle military software projects (Boehm, 1984).

The developer of the COCOMO model took a rather unorthodox view of his product in that he made public the algorithms upon which the model was based by documenting and publishing his research in a book, Software Engineering Economics (Boehm, 1981). Boehm's primary motivation for COCOMO was to help people understand what software cost models estimate and the consequences of decisions software managers make. COCOMO consisted of three increasingly detailed models--Basic, Intermediate and Detailed (Boehm, 1984). For all three versions, certain assumptions are made:

(1) Estimates are in man-months (MM) of direct labor required from the start of preliminary design to the end of acceptance testing.

(2) The primary driver is the number of source lines of code (SLOC) expressed as thousands of delivered source instructions (KDSI).

(3) There are no substantial changes in requirements (Ferens, 1994).

The basic COCOMO is useful for quick "ball park" estimates, while the intermediate and detailed versions are useful for more refined estimates. Basic COCOMO estimates effort based solely on program size in KDSI. Intermediate COCOMO improves upon the basic estimate by using 15 attributes, describing personnel capabilities, tools used, system requirements, etc., as additional cost drivers, to compute effort. The primary difference between the detailed and intermediate COCOMO is that the detailed COCOMO considers the phase sensitivity of the attribute ratings and values (Boehm, 1984). A summary of the basic and intermediate algorithms used for estimating effort are provided in Table 2-1 below.

TABLE 2-1 COCOMO Development Effort Algorithms

Mode	Basic Model	Intermediate Model
Organic	$MM = 2.4\ (KDSI)^{1.05}$	$MM = 3.2\ (KDSI)^{1.05}$
Semidetached	$MM = 3.0\ (KDSI)^{1.12}$	$MM = 3.0\ (KDSI)^{1.12}$
Embedded	$MM = 3.6\ (KDSI)^{1.20}$	$MM = 2.8\ (KDSI)^{1.20}$

COCOMO quickly became popular because it was not proprietary; it was free; and it was relatively easy to learn and operate. Needless to say, shortly thereafter a number of COCOMO variants began to appear on the market. One of these was REVIC.

More recently developed cost models have included models such as REVIC, SASET, and SEER-SEM. REVIC was developed by Ray Kile, an Air Force reserve officer (Kile, 1991), for use by U.S. Government and industry and is managed by the Air Force Cost Analysis Agency (AFCAA). The REVIC model was built using regression techniques and a database of 281 completed contracts with software involvement at the Rome Air Development Center. REVIC implements only the intermediate version of COCOMO. It also contains different coefficients and uses a Program Evaluation and Review Technique (PERT) of sizing for new programs. REVIC has four new input parameters not contained in COCOMO--requirements volatility, required reusability, security, and a management reserve risk factor. The algorithms for REVIC are comparable to the COCOMO algorithms, except for the coefficient, and are summarized in Table 2-2.

Table 2-2 REVIC Nominal Intermediate Equations

Mode	Effort Equation	Schedule Equation
Organic	$MM = 3.464\ (KDSI)^{1.05}$	$M=3.650\ (MM)^{0.38}$
Semidetached	$MM = 3.970\ (KDSI)^{1.12}$	$M=3.800\ (MM)^{0.35}$
Embedded	$MM = 3.312\ (KDSI)^{1.20}$	$M=4.376\ (MM)^{0.32}$
Ada	$MM = 6.800\ (KDSI)^{0.94}$	$M=4.376\ (MM)^{0.32}$

The Software Architecture, Sizing, and Estimating Tool (SASET), another non-proprietary cost model, was developed by Martin Marietta (now Lockheed Martin) for Navy and Air Force cost centers. Originally, SASET was intended as a non-proprietary DoD-only model. Although the model considers numerous factors and contains an exhaustive calibration file, it has failed to gain favor with estimators because it is not an easy model to learn or to use. The developer's failure to implement upgrades has also impacted on SASET's usefulness.

One of the more popular models used by the Air Force is the System Estimation and Evaluation of Resources Software Estimation Model (SEER-SEM). SEER-SEM was developed by Galorath Associates in 1987. It has a multitude of inputs, uses DoD terminology and is compatible with different development methods--spiral, waterfall, prototype, and incremental.

Summary of Prior Research

Cost analysts and managers have long recognized that improvements were needed in the capabilities of existing cost models to estimate software costs accurately. As early as 1978, Captain Walker, an AFIT graduate student, attempted to develop a software model that could assist in evaluating the effects of "modern" programming practices on the life-cycle cost of software.

The cost model which Walker developed was similar to SLIM, a cost model developed by Putnam, in that it was a *macro* cost estimation model. Macro cost models assume that cost driver attributes are applied uniformly across the entire product (Boehm, 1981). This approach is only good for rough order magnitude estimates such as those conducted early in the acquisition of a system. Walker's model added an additional parameter in an attempt to more accurately model the support costs found in the left tail of the Rayleigh distribution of life-cycle costs.

At the time of his study, Captain Walker noted that data availability placed a most severe limitation upon the ability to develop a model that would predict software costs with any reasonable degree of accuracy. In his opinion, the primary cause of poor database availability was due to four factors:

(1) the lack of data collection practices,

(2) competition among contractors,

(3) errors in data collection, and

(4) lack of consistency among data sets (Walker, 1978).

Walker found his efforts hampered by an inability to find, in literature, examples of any life-cycle costing of software systems. He also observed that cost models were restricted to either the development phase or to the operations and support phase (Walker, 1978).

In summarizing his efforts, Walker identified six factors which he felt affected software costs: (1) requirements, (2) hardware, (3) sizing, (4) management, (5) software unique parameters such as application, language, support software tools, structural design, and modularity, and (6) personnel capabilities and experience. When developing a cost estimate, Walker suggested that a sensitivity analysis be used to identify the best distribution of factors that contribute to cost (Walker, 1978).

Walker's observations are supported by the findings of Thibodeau in his evaluation of a number of software cost estimating models (Thibodeau, 1981). Thibodeau's findings mirrored what Walker and other researchers had already concluded--that model performance is very much environment dependent and that data availability and quality are a major limiting factor in cost model development. Thibodeau was of the opinion that the best way to develop a software cost estimate was to use the simplest model structure and to calibrate the model's parameters to represent the development environment (Thibodeau, 1981). His evaluation confirms the importance of data definitions to the interpretation of model performance and supports the recommendation that model development activities be used as the basis for establishing data reporting requirements under software development contracts. Thibodeau strongly suggests that software data reporting become an integral part of the contracting process much as operating costs are now, and that items and formats be defined by the Air Force and provided routinely by the contractors (Thibodeau, 1981).

About the same time that Thibodeau was making his observations, Dr. Barry Boehm was authoring a book to introduce the public to COCOMO, a software cost estimating model he had developed in response to the demand for control of escalating software costs (Boehm, 1981). The book documented his research into software cost modeling and has become a classic in the field of software development. Besides the problems of missing data and clerical inaccuracies, Boehm found some of the more frequent sources of software data collection problems stemmed from:

(1) inconsistent definitions such as different definitions for "delivered" instructions,

(2) observational bias,

(3) differences in local vs. global frames of reference,

(4) averaging and size effects, and

(5) double counting (Boehm, 1984).

The first study on the accuracy of REVIC was noted by Daly, during a study of software schedule estimation, in 1989. He reports that REVIC was accurate to within 25 percent of the actuals less than 30 percent of the time. However, he further noted that accuracy could be improved to 70 percent when adjustments were made during the preliminary design stage (Daly, 1990).

A second study of note occurred in 1991 when Ourada performed a calibration and validation of four models, REVIC, SASET, SEER, and COSTMODL in one development environment, using half of a subset of ground-based military programs, and doing a comparison using the remaining 14 programs as another development environment. When compared to the other subset, the accuracy results were mediocre, being accurate only to within 25 percent less than 30 percent of the time, even though the model had been calibrated for the other half of the subset. Interestingly, the model was more accurate after calibration, 25 percent for 70 percent of the time, for a subset of unmanned space programs--a subset for which REVIC had not been calibrated! Ourada also noted that, for all parameters, the coefficient-only calibration was more accurate than the coefficient and exponent calibration (Ourada, 1991).

In summary, Ourada found that REVIC was good at estimating outside the environment of calibration, but not good at estimating inside the environment. He concluded that the models were highly inaccurate and very dependent upon the interpretation of the input parameters (Ourada, 1991).

The latest study of significance was conducted by Coggins and Russell in 1993. Their study attempted to examine four cost models, REVIC, SASET, PRICE-S, and SEER-SEM, and to identify the differences in the models and the impact these differences had on cost estimates. Coggins and Russell also attempted to normalize, or adjust, the models in an effort to obtain comparable estimates from the various models. Their research identified differences that existed between the models at nearly every level.

In the case of REVIC, they found that the developer had not included a Systems Requirements Analysis/Design development phase and that the model did not differentiate between CSCIs, CSCs, and CSUs. They also noted that, according to the model developer, REVIC was limited to an estimating range of 500 to 130,000 SLOC (Coggins & Russell, 1993).

Their conclusions were that, although it was not particularly difficult to identify differences between the models, the differences in definitions for model inputs, internal equations, and key assumptions were so dissimilar as to render objective normalization efforts virtually impossible (Coggins & Russell, 1993).

Summary

This chapter has provided background information needed to understand the importance and relevance of this research. It appears, thus far, that most research into the accuracy of software cost models has arrived at similar conclusions--lack of cost data makes attempts to estimate software development costs a highly inaccurate science. Perhaps Ferrentino was correct when he stated that estimation is an educated guess and that there is no method to accurately predict the time and manpower needed to develop a software system, and that "we can't make good estimates, but we can make estimates good" (Ferrentino, 1981.). This research hopes to prove otherwise.

III. Methodology

Chapter Overview

This chapter addresses the data and methodology which will be used to calibrate the REVIC software cost model. Since the proper use of any software model requires a thorough understanding of the model's assumptions, capabilities, and limitations, the methodology will address these parameters as appropriate when they impact on the decisions made regarding selection of data and method of analysis.

Software Database

REVIC was calibrated using the Air Force Space and Missile Systems Center (SMC) software database (SWDB). The SMC SWDB is a recently updated database, last updated in December 1994, for the specific purpose of improving the estimating capability at SMC and to be used for this calibration effort. The database contains 2,614 records of software development and maintenance data and has 76 fields of information for each record. The software development data is provided at the project, the computer software configuration item (CSCI), the computer software component (CSC) and the computer software unit (CSU) levels. (Stukes, 1994).

The primary data sources for the SMC database were:

(1) The Space Systems Cost Analysis Group (SSCAG) and its contributing non-government SSCAG member organizations (primarily defense contractors);

(2) The USAF Space and Missile Center (SMC), which included the Aerospace Corporation and various Program Offices;

(3) Other Government Agencies, including the Air Force, Navy, and Army Cost Analysis Centers, the Army Missile Command and the Naval Air Development Center; and

(4) over 250 select government and industry organizations interested in software development and maintenance cost estimating and management.

Data collection forms and dictionaries were provided to the interested organizations to aid in the normalization of data collected (Stukes, 1994).

All new data received had been previously screened and entered into the SWDB automated user base by MCR, who had sanitized all data so as to exclude any proprietary or competition sensitive information, such as company name and program, and to protect the anonymity of the source. MCR had also normalized the new data as to effort and size and stratified it using a matrix which matched software applications with software functions. Other criteria which MCR used to stratify, or group, the records included: platform, software level, operating environment, software application, software function, programming language, and confidence level (Stukes, 1994).

However, further steps were required to normalize the effort and size before the data could be entered into the REVIC model. First, because REVIC considers a manmonth to be 152 hours, all effort was first standardized to 152 hours. Next, since REVIC calculates new and revised effort differently, the DSI of each project was normalized by adjusting new and reused DSI to mirror the total DSI as calculated by REVIC, using the formula:

$$EDSI = ADSI * [(.4\ DM + .3\ CM + .3\ IM)/100], \qquad \text{(Eq. 3.1)}$$

where EDSI is the equivalent DSI, and

ADSI is the adapted DSI.

The ADSI were then multiplied by the percent of design modification (DM), code modification (CM), and retesting (IM) required. No common code was included in the total DSI thus calculated. Finally, because of the way REVIC estimates effort for the software development phases, the effort for each project had to be normalized to reflect the equivalent REVIC effort. This process is explained in more detail in the following section.

Cost Model

Since the proper use of any software model requires a thorough understanding of the model's assumptions, capabilities, and limitations, the first priority of the researcher was to become familiar with the REVIC cost model, its limitations and capabilities, and the options available for calibration of the model to a specific database. One such characteristic of REVIC which required immediate attention was the manner in which the model addressed the major phases which occur during the development process.

The typical software development process, as described in DoD Standard 2167A, consists of eight phases:

(1) The System Requirements Analysis and Design Phase,

(2) The CSCI Requirements Analysis Phase,

(3) The Preliminary Design Phase,

(4) The Detailed Design Phase,

(5) The Coding and CSU Testing Phase,

(6) The CSC Integration and Testing Phase,

(7) The CSCI Testing Phase, and

(8) The System Integration and Testing.

REVIC estimates costs for only six of the eight development phases identified above. REVIC initially calculates and allocates effort to four phases, Preliminary Design through CSCI Testing while combining two of the phases, Coding & CSU and CSC Integration and Testing (See Table 3-1). REVIC then adds 12% to the resulting development effort for the Software Requirements Analysis Phase and 22% for the Systems Test & Integration Phase. Normalization of the data was complicated because the SMC SWDB used different terminology for the eight phases. A further complication arose due to the different percentages which REVIC and the SMC SWDB assigned to each phase. First the five SMC SWDB core effort phases (preliminary design through CSCI test) were normalized to the four core REVIC phases (preliminary design through integration and test). Finally, the *core* normalized effort was adjusted to mirror the actual phases included in the effort. These phases are summarized in the second page of Appendices C and D. A summary of the phases and their percentage of effort allocated by REVIC and the SMC SWDB are provided below.

Table 3-1: A Comparison of Development Phase Terminology

DoD Mil-STD 2167-A	REVIC	SMC SWDB
Sys Reqrs Anal & Design Phase	None	None
CSCI Requirements Anal Phase	Software Requir Phase *(12%)*	SW Requirements Phase *(12%)*
Preliminary Design Phase	Preliminary Design Ph *(23%)*	Prelim Design Phase *(11.4%)*
Detailed Design Phase	Critical Design *(29%)*	Detail Design Phase *(19.1%)*
Coding & CSU Testing Phase	Code & Debug *(22%)*	Code & Unit Test Phase *(29.8%)*
CSC Integration & Testing	Code & Debug	CSC Test & Integr Ph *(35.6%)*
CSCI Testing Phase	Integration & Test Phase *(26%)*	CSCI Test Phase *(4.1%)*
System Integration & Testing	Dev Test & Integration Ph *(22%)*	Sys Test & Integr Phase *(7.2%)*
None	None	Opn Test & Eval Phase *(4.8%)*

Please note that in REVIC, the core phases (preliminary design through integration and test) total 100% as does the SMC SWDB core phases (preliminary design through CSCI test). Yet when the additional phases are included, the percentages become a total of 134% for REVIC and 117.5% for the SMC SWDB. Needless to say, normalizing the SMC SWDB effort to REVIC equivalent effort developed into a real chore.

A considerable amount of time was also spent becoming familiar with the nineteen parameters which REVIC uses to arrive at a complexity factor (Π or EAF) for each project and determining their equivalent parameters identified in the SMC SWDB. A summary of this comparison is included in Appendices C and D. The REVIC parameters are summarized in Table 3-2, on the next page.

Table 3-2: Key to REVIC Parameters

Parameters	Description
ACAP	Analyst's Capability
PCAP	Programmer's Capability
AEXP	Applications Experience
VEXP	Virtual Machine Experience
LEXP	Language Experience
TIME	Processing or Throughput Constraints
STOR	Storage/Memory Constraints
VIRT	Virtual Machine Volatility
TURN	Turnaround Time
RVOL	Requirements Volatility
RELY	Required Reliability
DATA	Data Base Size
CPLX	Code Complexity
RUSE	Required Reusability
MODP	Modern Programming Practices
TOOL	Use of Design and Programming Tools
SECU	DoD Security Classification
RISK	Risk associated with Platform
SCED	Schedule Compression/Stretch Out

An examination of the SMC database revealed a lack of information for all key parameters. For this reason, a determination was made to limit the multipliers in each operating environment (i.e. military ground and unmanned space) to those parameters which contained complete information for all data points selected. The parameters used for each operating environment are also summarized in the worksheets in the Appendices. In essence, this defaults the other parameters to a value of "1". Because of the manner in which REVIC considers the phases, and the adjustments which this entailed, the data were also examined for completeness of information regarding the phases since this information was essential in determining the adjusted effort.

Method of Stratification

Prior to any stratification efforts, this researcher decided that more than eight records would be required before calibration would be attempted on any data set, with stratification being by operating environment, as requested by the sponsor. Initial stratification resulted in the identification of two operating environments with sufficient records for calibration: Military Ground, and Unmanned Space. Within each platform, data points were stratified by software level. Since REVIC does not differentiate between CSCI's CSC's, and CSU's, only data at the CSCI software level was identified. Data points were further limited to U.S. only efforts. All software applications and functions were included. Excluded from the database were those programs which consisted of Assembly, Machine, and Microcode language. Since the recommended estimating range for REVIC is from approximately 500 to 130,000 SLOC per REVIC data file, (Coggins & Russell, 1993), only data points within this range were selected. During selection of data points, it was also discovered that using the category of confidence level and limiting the search to those records with a *nominal* to *high* confidence level speeded the screening out of those records with incomplete information. In other words, *confidence level* was viewed as a way to rank the completeness of information available on a particular record.

Method of Analysis

The principal method of analysis was regression analysis and included, first, the technique of linear least squares best fit following the procedures recommended by Boehm (Boehm,1981). This technique was applied to entire data sets as well as to subsets of data. Error reduction in the model's predicting ability was examined in terms of the magnitude of the relative error (MRE), where

$$MRE = |Y_{actual} - Y_{predicted}| / |Y_{actual}|, \qquad \text{(Eq. 3.2)}$$

the mean magnitude of relative error (MMRE), where

$$MMRE = 1/n * \sum MRE_i , \qquad \text{(Eq. 3.3)}$$

the root mean square error (RMS), where

$$RMS = [\ 1/n \sum(Y_{actual} - Y_{predicted})^2\ ]^{1/2}, \text{ and} \qquad \text{(Eq. 3.4)}$$

the relative root mean square (RRMS) error, where

$$RRMS = RMS/\ (1/n \sum Y_{actual}). \qquad \text{(Eq. 3.5)}$$

A final statistical test which was used was

$$PRED\ (.30) = k/n, \qquad \text{(Eq. 3.6)}$$

a prediction level test where k is the number of projects in a set of n projects whose MRE is less than or equal to 30% (Conte, 1986).

A second analysis was also conducted using a standard statistical software package, in this case SAS®, to arrive at an algorithm. A comparison of the results of the two methods of calibration was then made to determine if the two methods produced similar results. The results should provide some insight into the importance of the role of the REVIC parameters which are used as constant multipliers in the REVIC model.

Calibration was limited to calibration of operating environments. As stated earlier, two operating environments were used for this study. The first environment from which data points were selected was the Military Ground platform. This platform was selected to be calibrated first because it contained a larger database and appeared to have the potential for a greater number of data points. The second environment selected for calibration was the Unmanned Space platform. Unmanned Space contained one of the smaller data bases of those platforms examined. A third operating environment, Military Mobile, was considered, but yielded only eight data points. Such a small data set did not permit the use of selected data points to be used as controls according to the guidelines established by the sponsor. For this reason, no calibration was attempted for the Military Mobile operating environment. All other operating environments in the SMC SWDB yielded fewer than eight data points.

Values for the coefficients and exponents in the REVIC algorithm were determined and tested using the steps requested by the sponsor:

(1) Select data points for calibration from a homogeneous data subset. This was the most difficult part of the study. Although MCR had provided sources with a data collection form containing guidance as to the information needed and had conducted an extensive effort to locate critical missing pieces of information, the data still contained much missing information. Each data point was scrutinized for completeness of information before being included in the data set. As a result, of the total 1,614 records in Military Ground, only 11 were determined to meet the requirements for completeness. Of the total 206 records in Unmanned Space, only 13 were found to meet the requirements for completeness.

(2) Set aside several data points from those data identified to be used for validation. Those projects chosen as controls were selected at random using a method requested by the sponsor and thesis advisor. First all data points were listed in order of size. Then, after selecting a "seed" project at random, every third data point from the "seed" was selected as a control project until a predetermined number of controls had been identified. The total number of projects to be retained as controls was determined using the following criteria:

(a) If total data points total 8 or less, use all points to calibrate;

(b) If total data points total 9 to 12 points, use 8 to calibrate and the others as controls to validate improved estimating capability of model;

(c) If total data points total more than 12 points, use 2/3 of the points to calibrate and 1/3 of the points as controls.

(3) Adjust effort for REVIC capabilities.

(4) Determine the predicted costs of each data point using the REVIC cost model before any calibrations are conducted.

(5) Using the larger data set, adjust the REVIC model parameters, using linear regression techniques, to a least squares best fit algorithm from the known data.

Adjustment will be made to the model coefficients only, and to both the model coefficient and exponent using the techniques suggested in the REVIC User's Guide (Kile, 1991) and by Dr. Boehm (Boehm, 1981).

(6) After the model has been calibrated, once again predict costs of the control group and compare those predicted costs with predicted costs obtained from the uncalibrated model.

Finally, the results were examined to ensure that the basic assumptions of regression analysis held. Methods of analysis used for this examination were the Wilcoxon Signed Rank Test and the Wilk-Shapiro/Rankit Plot of Residuals. The *Wilcoxon Signed Rank Test* is a nonparametric alternative to the Paired-T Test and requires virtually no assumptions about the paired samples other than that they are random and independent. The Wilcoxon Signed Rank Test assumes that you have two groups and have drawn samples in pairs. It tests the hypothesis that the frequency distribution for the two groups are identical (Mendenhall et al, 1990).

The *Wilk-Shapiro/Rankit Plot of Residuals* is useful for examining whether the test assumptions in regression have been violated by examining whether a variable conforms to a normal distribution. A rankit plot of the variable is produced and an approximate Wilk-Shapiro normality statistic (Shapiro-Francia) is calculated (Sieget, 1992).

Method of Calibration

Using Boehm's methodology for a coefficient only calibration:

(1) Determine the most appropriate constant, "***c***", for the nominal effort equation in the REVIC estimating relationship,

$$MM = c(KDSI)^{1.20}\,\Pi\,(EM), \qquad \text{(Eq. 3.7)}$$

where Π (EM) represents the overall product of the effort multipliers resulting from a project's cost driver attribute ratings, or more concisely, its effort adjustment factor, Π.

(2) Solve for the value of "*c*" in the system of linear equations,

$$MM = c(KDSI)^{1.20}\,\Pi_1 \ldots$$

$$\Downarrow$$

$$\ldots\; MM = c(KDSI)^{1.20}\,\Pi_n \qquad \text{(Eq. 3.8)}$$

which minimized the sum of the squares of the residual errors

$$S = \Sigma\,[c\,(KDSI)^{1.20}\,\Pi_i - MM_i\,]^2\text{, and}$$

setting $(KDSI)^{1.20}\,\Pi_i = Q_i$ for simplicity, the equation becomes

$$S = \Sigma\,[c\,Q_i - MM_i\,]^2. \qquad \text{(Eq. 3.9)}$$

(3) We can then determine the optimal coefficient "$\mathbf{c_{mean}}$" by setting the derivative dS/dc equal to zero and solving for the mean of "**c**",

$$0 = dS/dc = 2\,\Sigma\,[c_{mean}\,Q_i - MM_i\,]\,Q_i\text{ , or}$$

$$0 = \Sigma\,c_{mean}\,Q_i^2 - MM_i\,Q_i\,.$$

Thus the mean of "**c**" becomes

$$c_{mean} = \Sigma\,MM_i\,Q_i\,/\,\Sigma Q_i^2, \qquad \text{(Eq. 3.10)}$$

using the form in Table 3-3,

Table 3-3. Calibrating the Constant Term

Project	$KDSI_i$	Π_i	MM_{est}	MM_i	Q_i	MM_iQ_i	Q_i^2

where Π_i is the Effort Adjust Factor for n = 1, 2, ...n;

MM_{est} is the effort estimated by the uncalibrated model for n = 1, 2, ...n;

MM_i is the actual REVIC equivalent effort; and

Q_i is equal to $(KDSI)^{1.20}$ for n = 1, 2, ...n.

A similar least-squares technique may be used <u>to calibrate both the coefficient term *c* and the exponent factor *b*</u> in the REVIC effort equation:

(1) First we rearrange the equation

$$MM = c(KDSI)^b\,\Pi\,(EM)\text{, to}$$

$$c(KDSI)^b = MM/\Pi. \qquad \text{(Eq. 3.11)}$$

(2) Then we make the equation linear by taking the logarithm of both sides so that we have

$$\log c + b \log (KDSI) = \log (MM/\Pi). \qquad \text{(Eq. 3.12)}$$

(3) Our next step will be to solve for the values of **log *c*** and for ***b*** so as to minimize the sum of the squares of the residual errors. We do this by solving the equations

$$a_0 \log c_{mean} + a_1 b_{mean} = d_0, \text{ and} \qquad \text{(Eq. 3.13)}$$

$$a_1 \log c_{mean} + a_2 b_{mean} = d_1 \qquad \text{(Eq. 3.14)}$$

where the quantities a_0, a_1, a_2, d_0, and d_1 are calculated as:

$$a_0 = n \qquad \text{(Eq. 3.15)}$$

$$a_1 = \Sigma \log (KDSI)_i \qquad \text{(Eq. 3.16)}$$

$$a_2 = \Sigma [\log (KDSI)_i]^2 \qquad \text{(Eq. 3.17)}$$

$$d_0 = \Sigma \log (MM/\Pi)_i \qquad \text{(Eq. 3.18)}$$

$$d_1 = \Sigma \log (MM/\Pi)_i \log (KDSI)_i. \qquad \text{(Eq. 3.19)}$$

(4) The solutions above are then used to find $\log c_{mean}$ and b_{mean}, and we have:

$$\log c_{mean} = (a_2d_0 - a_1d_1) / (a_0a_2 - a_1^2), \text{ and} \qquad \text{(Eq. 3.20)}$$

$$b_{mean} = (a_0d_1 - a_1d_0) / (a_0a_2 - a_1^2) \text{ (Boehm, 1981).} \qquad \text{(Eq. 3.21)}$$

Finally, a similar analysis will be conducted using the SAS® statistical software package and standard statistical analysis procedures. Since the REVIC algorithm is a multiplicative cost estimating equation of the form $Y_{predicted} = B_0*X^{B1}$, to derive a multiplicative cost estimating equation will require three steps:

First, we take the logarithm of the X and Y, so that $Y_{predicted} = B_0*X^{B1}$ becomes a linear model,

$$\log (Y) = \log (B_0) + B_1 \log (X). \qquad \text{(Eq. 3.22)}$$

Next, we derive the linear least squares best fit equation in terms of the logarithms of Y, X, and B_0.

Finally, we transform this equation back into the X and Y space.

A comparison of the results will be made to determine the differences obtained, if any, using the two methods. The results of the SAS® analysis will also be used to evaluate the basic assumptions inherent in the least squares best fit methodology of analysis.

Summary

This chapter has reviewed the data that will be used for this research, the methodology that will be used to select and analyze data points to be used in the calibration, and the statistical techniques to be used to perform the calibration, validation, and comparison.

IV. Analysis and Findings

Chapter Overview

This chapter presents the analysis and findings of the calibration effort on REVIC. The analysis of the SWDB begins with a normalization of SLOC for input into REVIC. Results of the original estimates, before calibration are given. The calculations to calibrate REVIC are made and the resulting estimates after calibration are compared with the original estimates. The resulting cost estimating relationship (CER) obtained for each operating environment, as a result of the calibration, is provided.

Military Ground

The REVIC algorithm was calibrated to the Military Ground operating environment using eight of the eleven projects listed in Table 4-1. Projects used to calibrate the model had a mean of 408.1 MM with a standard deviation of 233.3 MM. The projects chosen at random as controls, and *not included* in the calibration of REVIC, were project numbers 2517, 2610, and 2612. The controls were used to measure the change in REVIC's estimating accuracy after calibration.

Table 4-1: Military Ground Calibration

Project No.	$KDSI_i$	Π_i	MM_{est}	MM_i	Q_i	MM_iQ_i	Q_i^2
2497	10.000	1.126	66..2	89.4	17.85	1,595.42	318.48
2501	106.200	1.324	1,586.1	542.6	357.47	193,962.24	127,783.37
2510	43.437	0.895	306.7	193.6	82.65	16,001.48	6,831.35
2517	**90.000**	**0.697**	**684.4**	**235.3**	**154.28**	**36,302.08**	**23,803.86**
2521	97.087	0.486	522.5	954.0	117.82	112,403.01	13,882.26
2526	6.681	0.838	27.0	208.8	8.19	1,709.17	67.01
2527	7.457	0.838	30.6	232.5	9.34	2,171.42	87.73
2528	21.588	0.838	109.7	673.8	33.44	22,533.57	1,118.43
2610	**14.536**	**0.838**	**68.3**	**453.7**	**20.81**	**9,441.50**	**432.85**
2611	11.840	0.838	53.4	370.0	16.27	6,018.22	264.55
2612	**9.899**	**0.838**	**43.1**	**309.0**	**13.12**	**4,054.39**	**172.16**
Total						356,394.52	150,352.81

Calibration produced the following algorithms:

(1) Calibration of the coefficient only.

$$MM = 2.370\ (KDSI)^{1.20}\ (\prod) \qquad \text{(Eq. 4.1)}$$

(2) Calibration of the coefficient and exponent.

$$MM = (84.868)\ (KDSI)^{0.474}\ (\prod). \qquad \text{(Eq. 4.2)}$$

(3) Calibration using SAS and ignoring the effort adjustment factors.

$$MM = (81.2126)\ (KDSI)^{0.4424} \qquad \text{(Eq. 4.3)}$$

Note the near similar results obtained for equations 4.2 and 4.3. A detailed account of the calculations and methodologies can be found in the military ground worksheets in Appendix C.

Using the predicted values for effort obtained from each of the three equations above (Eqs. 4.1, 4.2, 4.3), the MRE, MMRE, RMS, and RRMS were calculated. Although all four measures look at the estimating error in different ways, in all instances, a smaller value means that, for that data point or for that control group, the model did a better job of predicting the actual effort. A summary of the changes produced by calibration are noted in Tables 4-2 and 4-3. Results were mixed with no one calibration method consistently producing superior results. Looking first at the MRE, no conclusions could be drawn (Table 4-2).

Table 4-2: Calibration Effects on Military Ground MRE

Project No.	Prior to Calibration	Calib. of Coeff.	Calib. of Coeff & Exp.	Calib. Using SAS
2517	1.91	1.30	1.80	2.38
2610	0.85	0.89	0.45	0.26
2612	0.86	0.90	0.33	0.08

Since the MMRE is more meaningful than the MRE, this was the next statistic to be examined. For the model to be acceptable as an estimating tool, the MMRE should be less than, or equal to 0.25. (Conte, Dunsmore & Shen, 1986). Obviously, if one looks at the MMRE, calibration did not sufficiently improve the model so as to make REVIC a useful model for estimating military ground software development. Table 4-3 summarized the effect of the calibration on the MMRE, and on three other statistics, the Root Mean

Square Error (RMS), the Relative Root Mean Square Error (RRMS), and the prediction level test (PRED).

The RMS represents the mean value of the error minimized by the regression model. From the RMS, we obtain the RRMS. (Conte & all, 1986).

Table 4-3: Calibration Results on Military Ground Estimates

Statistical Test	Prior to Calib.	Coefficient Calib.	Coeff. & Exp. Calib.	SAS Calib.
MMRE	1.21	1.03	0.86	0.91
RMS	374.58	332.62	227.10	331.17
RRMS	1.13	1.00	0.68	1.00
PRED (.25)	0%	0%	0%	33%

Unfortunately, the first four criteria, MRE, MMRE, RMS, and RRMS, are often not in agreement. In a situation where the criteria do not agree, no determination can be made as to which model is best except by making a subjective judgment on the relative importance of the individual evaluation criteria. In this case, one might want to select the model which makes predictions that have the smaller average errors (Conte et al, 1986).

In general, it appears that calibration resulted in some improvement in REVIC's estimating ability in all instances. However, the simultaneous calibration of the coefficient and exponent, using the effort adjustment factor (EAF) as a constant multiplier (Boehm's methodology) appears to have provided the most improvement. None of the calibration efforts produced a model with the desired estimating accuracy. The prediction level test (PRED) at the 25% level (Table 4-3) reveals that 0% of the predicted values fell within 25% of their actual values for the coefficient only and for the simultaneous coefficient and exponent calibration. Only the calibration using SAS®, and ignoring the constant multipliers, resulted in an improvement with 33% of the predicted values falling within 25% of their actuals. A further examination of the scatter plot and residuals of the sample data used to calibrate REVIC provides further insight (Figures 4-1 and 4-2).

In Figure 4-1, it appears that we may have two distinct and separate relationships being modeled by the data with projects 5,6,7, 8, and (perhaps) 4 representing one relationship and projects 1, 2 , and 3 modeling another relationship. An examination of the residuals (Figure 4-2) reinforces this suggestion. If this should be the case, no useful relationship can be obtained using the least squares best fit (LSBF) methodology within the constraints of the REVIC algorithm because the algorithm is limited to a single independent variable (SIV). This would make REVIC inappropriate for predicting the effort for development projects with multiple independent variables (MIV), such as the military ground environment appears to have.

Military Ground

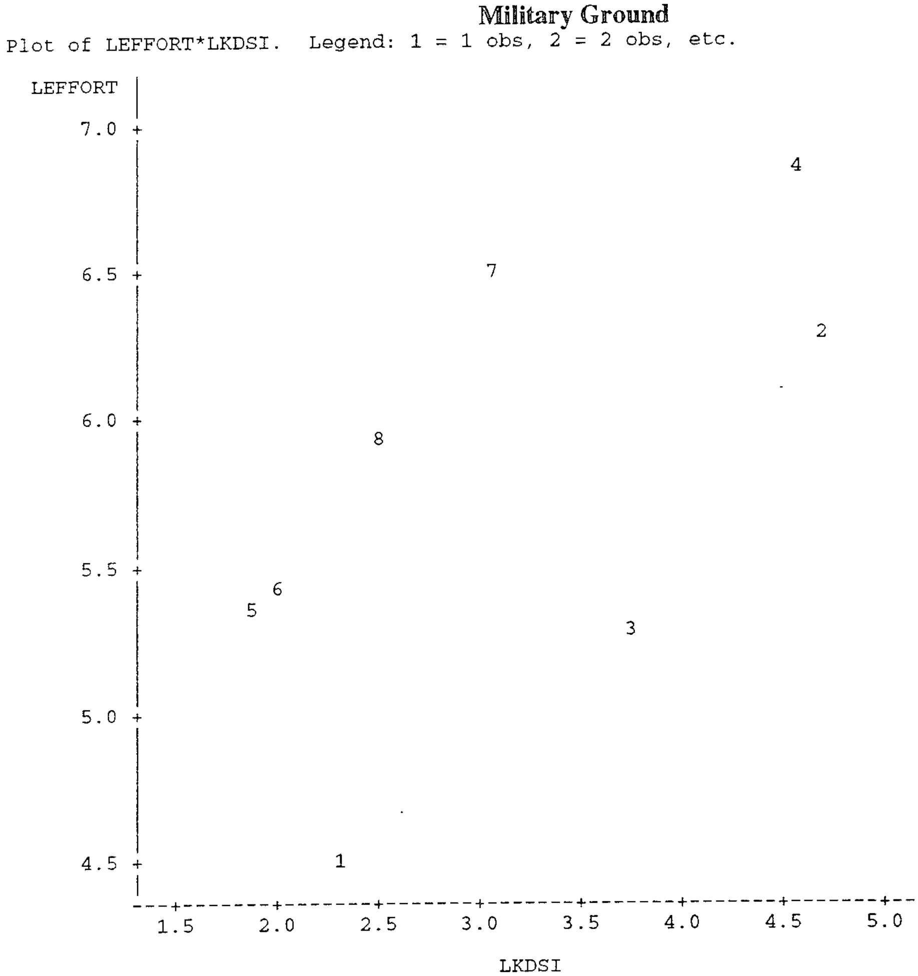


Figure 4-1. Military Ground Scatter Plot

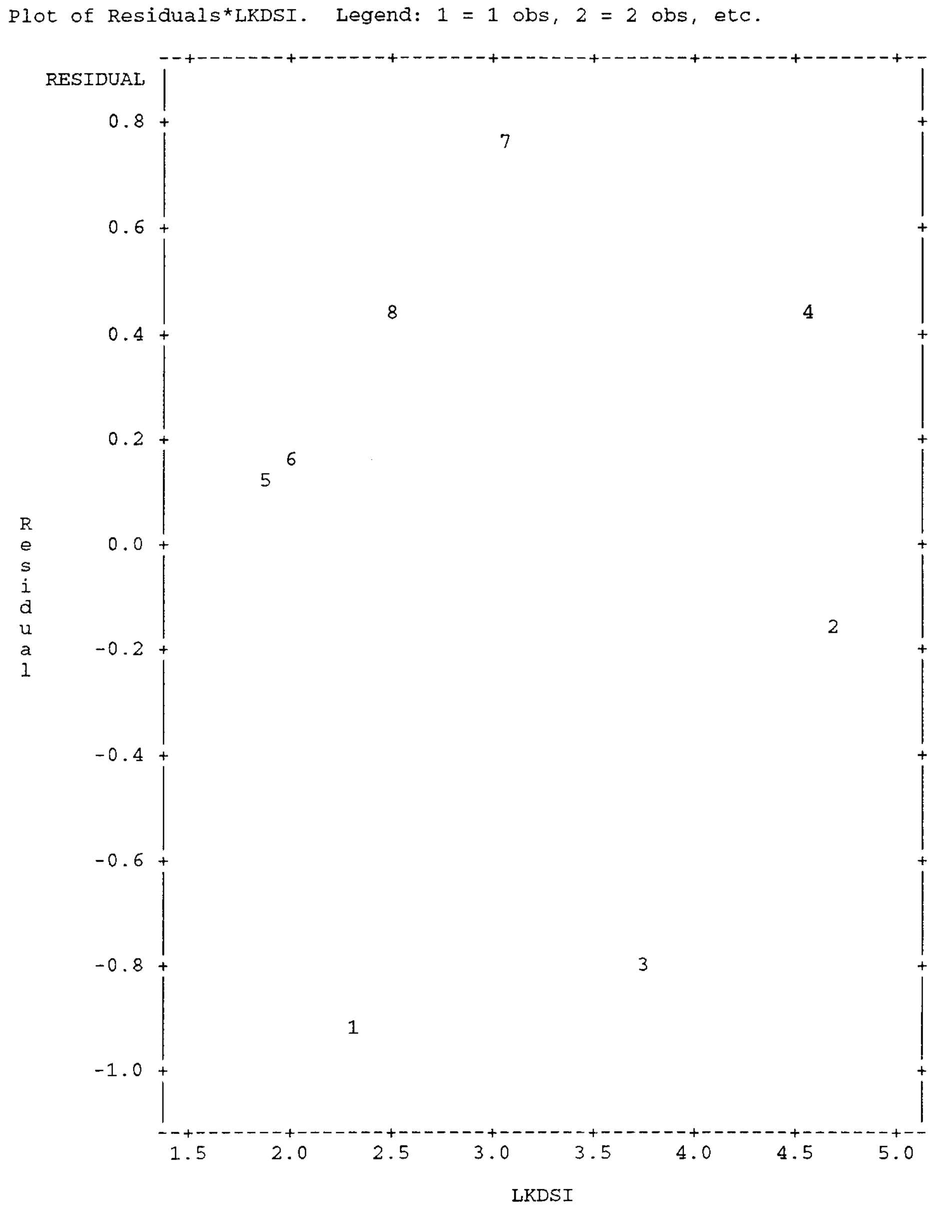


Figure 4-2. Military Ground Residuals

The Wilcoxon Signed-Rank test was carried out on all three--the coefficient only, the coefficient and exponent, and the SAS®calibrations, to test the hypothesis that the relative frequency distributions resulting from each calibration was identical to the actual distribution. Because the amount of data was small, the test was conducted using α = 0.10, which resulted in the critical value of T (T_{crit}) = 6. Therefore, if the calculated value of T (T_{calc}) proved to be less than or equal to 6, the hypothesis that the relative frequency distributions of the two populations were identical could be rejected. Obviously, the hypothesis that the distributions were identical could not be rejected.

Table 4-4: Military Ground Wilcoxon Signed Rank Tests

	Pre-Calibration	Coeff Only	Coeff & Exp	SAS Log-Log
$T_{critical}$	6	6	6	6
$T_{calculated}$	10	8	17	14

Finally, the assumptions of linear regression were examined. If the assumptions are met, the residuals should be approximately normally distributed with a mean of zero ($\mu = 0$) and a variance of one ($\sigma^2 = 1$). To do this, the Wilk-Shapiro/Rankit Plot of Residuals was used. First the order statistics of the sample were determined. This was done by reordering sample values by their rank. If the residuals are normally distributed, the plot of rankits against the ordered statistics should result in a straight line except for random variation. A systematic departure of the rankit plot from a linear trend indicates non-normality, as does the small value for the Wilk-Shapiro statistics. One, or a few points, departing from the linear trend near the extremes of the plot are indicative of outliers. Note that the normality plot in Figure 4-3 shows both asterisks (*) and plus signs (+). The plus signs from a straight line. The asterisk signs represent the sample. If the sample is from a normal distribution, the asterisks form a straight line and thus cover most of the plus signs. As can be seen from the Military Ground Rankit Plot of residuals, in Figure 4-3, most of the asterisks in the plot for Military Ground cover the plus signs. Therefore, we can conclude that the residuals are normally distributed and the assumptions of linear regression are met by the Military Ground data set. This conclusion is further reinforced by additional tests for normality. Looking at the outputs in Appendix C, page C-1, the bottom line of the *Moments* table shows the results for normality. The column labeled *W:Normal* gives the value of the test statistic. The test statistic, *W*, is greater than zero and less than or equal to one. Values of *W* that are too small indicate that the data are not a sample from a normal distribution. The second column, labeled *Prob < W*, contains the probability value, which describes how doubtful the idea of normality is. Probability values (p-values) can range from zero to one. Values very close to zero indicate the data are not a sample from a normal distribution and produce the most doubt. (Schlotzhauer & Littell, 1987). For the Military Ground, this researcher concluded the data are normally distributed.

```
                          Univariate Procedure
                           Variable=Residuals

                             Normal Probability Plot
   350+                                                *   ++*++
      |                                              +++++
      |                                          ++*+
    50+                                     *+++*
      |                                ++*++
      |                            ++*+
  -250+                       ++*++
       +----+----+----+----+----+----+----+----+----+----+
           -2        -1         0        +1        +2
```

Figure 4-3: Military Ground Normality Test

Unmanned Space

For the second calibration effort, the REVIC algorithm was calibrated to the Unmanned Space operating environment using nine of the thirteen projects listed in Table 4-5. Projects used to calibrate the model had a mean of 263.19 MM with a standard deviation of 100.34 MM. Projects chosen at random as controls, and *not included* in the calibration of REVIC, were Project numbers 77, 78, 82, and 306. The controls were used to measure the change in REVIC's estimating accuracy after calibration.

Calibration produced the following algorithms:

(1) Calibration of the coefficient only.

$$MM = 1.5274\,(KDSI)^{1.20}\,(\Pi) \qquad \text{(Eq. 4.4)}$$

Table 4-5: Unmanned Space Calibration

Project No.	$KDSI_i$	Π_i	MM_{est}	MM_i	Q_i	MM_iQ_i	Q_i^2
74	11.700	1.366	86.6	80.0	26.14	2091.05	683.20
75	116.800	2.131	2,135.6	912.0	644.93	588,179.25	415,939.08
76	14.000	1.392	109.5	115.0	33.04	3799.19	1,091.40
77	**56.200**	**2.131**	**887.7**	**523.0**	**268.08**	**140,205.84**	**71,866.89**
78	**48.300**	**1.188**	**412.5**	**478.0**	**124.61**	**59,563.58**	**15,527.65**
79	50.300	1.188	433.1	432.0	130.83	56,517.35	17,115.75
80	69.450	1.188	637.9	296.0	192.67	57,031.51	37,123.27
81	22.900	1.188	168.5	164.0	50.89	8,345.70	2,589.63
82	**16.300**	**1.188**	**112.1**	**140.0**	**33.84**	**4,737.6**	**1,145.15**
83	6.800	1.009	33.4	57.0	10.07	573.82	101.35
306	**9.400**	**0.566**	**20.4**	**69.4**	**8.33**	**578.10**	**4,815.57**
2516	48.814	0.684	269.5	197.5	72.66	14,347.91	5,279.82
2518	18.004	1.001	142.5	115.2	32.13	3702.37	1,032.18
Total						734,588.14	480,955.68

(2) Calibration of the coefficient and exponent.

$$MM = 10.8489\ (KDSI)^{0.800}\ (\Pi) \qquad \text{(Eq. 4.5)}$$

(3) Calibration using SAS® and ignoring the effort adjustment factors.

$$MM = 9.6888\ KDSI^{0.8859} \qquad \text{(Eq. 4.6)}$$

Here again, the algorithms between the second and third calibration produced coefficients and exponents with similar results (Eqs. 4.5 and 4.6), implying that the effort adjustment factor (EAF) may not be the major cost driver it is thought to be. A summary of the calculations and methodologies for the unmanned space operating environment can be found in Appendix D.

In the same manner of analysis used for the military ground environment, the MRE, MMRE, RMS, RRMS, and PRED were calculated. A summary of the changes produced in the MRE, by each calibration, is noted in Table 4-6.

Table 4-6: Calibration Effects on Unmanned Space MRE

Project No.	Prior to Calibration	Calib. of Coeff.	Calib. of Coeff & Exp.	Calib. Using SAS
77	0.6973	0.2172	0.1098	0.3426
78	0.1370	0.6017	0.4004	0.3709
82	0.1993	0.6307	0.1421	0.1793
306	0.7061	0.8646	0.6066	0.0159

The effect of calibration on the mean magnitude of relative error (MMRE), the Root Mean Square Error (RMS), the Relative Root Mean Square Error (RRMS), and the prediction level test are summarized in Table 4-7.

Table 4-7: Calibration Results on Unmanned Space Estimates

Statistical Test	Prior to Calib.	Coefficient Calib.	Coeff. & Exp. Calib.	SAS Calib.
MMRE	0.435	0.579	0.315	0.227
RMS	187.400	163.566	102.588	126.668
RRMS	0.619	0.541	0.339	0.4186
PRED (.25)	50%	25%	50%	50%

Keeping in mind that, in all cases, a smaller value means better predicting, once again results were mixed. Based on the MMRE, which is more meaningful than the MRE, only the SAS®calibration of coefficient and exponent, which excluded the EAF as a constant cost multiplier, produced a model with an acceptable estimating accuracy. If one focuses on the model with the smaller mean values of errors, it appears that the model produced using Boehm's methodology for simultaneous calibration of the coefficient and exponent produced the best model. However, since the RRMS is greater than 0.25 in all cases, none of the calibration attempts produced an acceptable model. Based on the prediction level test, in no instance did calibration improve the model's estimating ability. Fifty percent of the predicted values were falling within 25% of their actuals before the model was calibrated. Calibration did not improve upon the predictions; in fact, the *coefficient only* calibration actually made the model predict less accurately!

Results obtained from attempts to calibrate to the unmanned space operating environment were especially disappointing because this effort was expected to be more successful than the attempt to calibrate to the Military Ground operating environment. An examination of the scatter plot, when the log of effort is plotted against the log of KDSI, reveals a more homogeneous data set with a very well defined relationship, as can be seen by examining Figures 4-4.

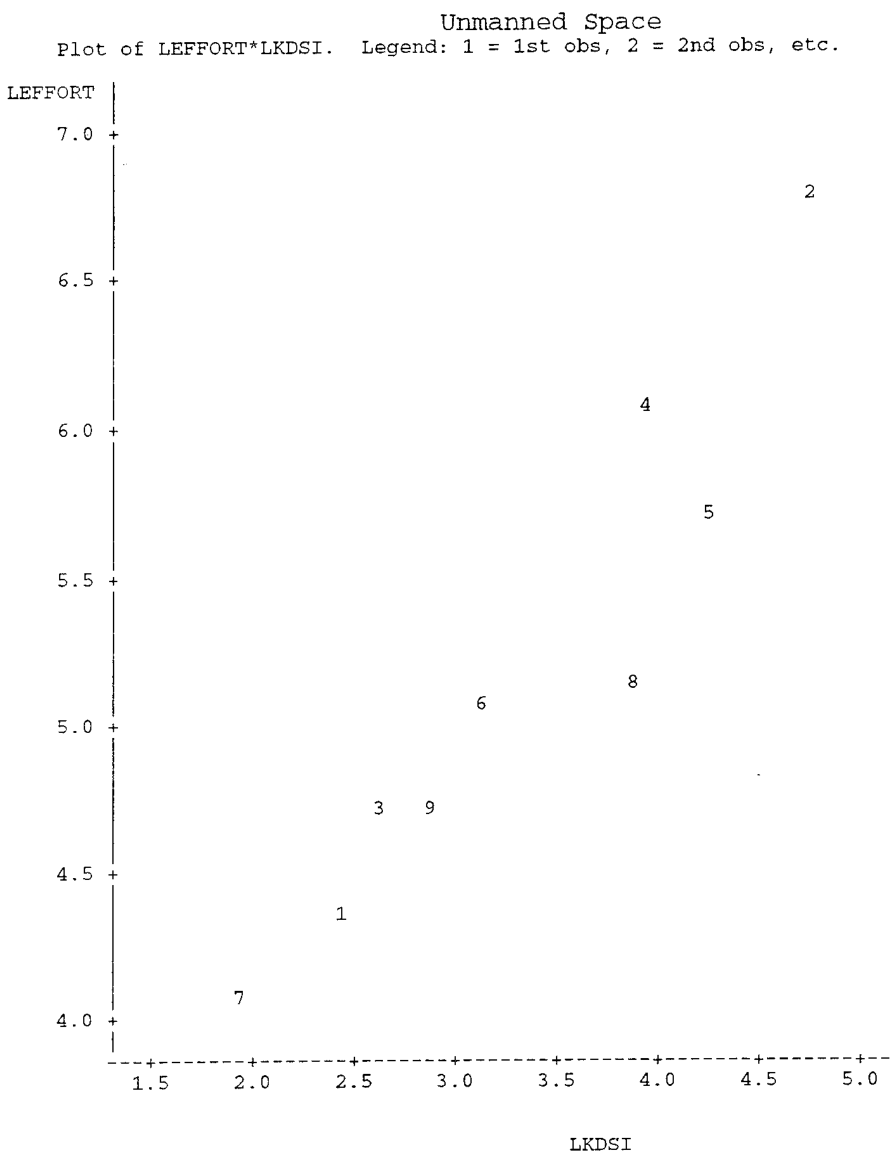


Figure 4-4. Unmanned Space Scatter Plot

An examination of the residual plot, in Figure 4-5, provides a clue as to why the calibration did not produce better results. Based on the residuals, it appears that the data set is highly heteroscedastic with the errors becoming greater as the size of the program developed becomes larger.

```
        Plot of Residuals*LKDSI.  Legend: 1 = 1st obs, 2 = 2nd obs, etc.

            -----+------+------+------+------+------+------+------+------
   RESIDUAL |                                                           |
        0.4 +                                                           +
            |                                                           |
            |                                                           |
            |                                                           |
            |                                      4           2        |
        0.3 +                                                           +
            |                                                           |
            |                                                           |
            |                                                           |
            |                                                           |
        0.2 +                                                           +
            |                                                           |
            |                                                           |
            |                    3                                      |
            |                                                           |
        0.1 +                                                           +
            |          7                                                |
            |                           6                               |
            |                                                           |
R           |                                                           |
e       0.0 +                                                           +
s           |                                                           |
i           |                                                           |
d           |                 1                                         |
u           |                       9                                   |
a      -0.1 +                                                           +
l           |                                                           |
            |                                                           |
            |                                                           |
            |                                                           |
       -0.2 +                                                           +
            |                                                           |
            |                                                           |
            |                                                           |
            |                                                           |
       -0.3 +                                                           +
            |                                                           |
            |                                          5                |
            |                                                           |
            |                                                           |
       -0.4 +                                                           +
            |                                                           |
            |                                     8                     |
            |                                                           |
            |                                                           |
       -0.5 +                                                           +
            |                                                           |
            |                                                           |
            -----+------+------+------+------+------+------+------+------
                1.5    2.0    2.5    3.0    3.5    4.0    4.5    5.0

                                        LKDSI
```

Figure 4-5. Unmanned Space Residuals

Interestingly, information received from MCR, following the calibration of REVIC to the Unmanned Space operating environment, revealed that the database was in error and that the Unmanned Space database actually contained data for two operating environments--Unmanned Space and Military Ground in support of Unmanned Space. Most of the projects selected from Unmanned Space to calibrate REVIC were actually not unmanned space projects. Only two of the projects initially identified as unmanned space projects, projects 306 and 2518, were correctly identified. All other projects were ground based projects in support of unmanned space programs.

In light of this knowledge, it is interesting to note that when project 306 was used as a control project to validate the calibration effort, it produced the largest magnitude of relative error in all instances except for the calibration using SAS®, producing in that instance, the smallest MRE value of all the control projects.

The Wilcoxon Signed-Rank test was again carried out on all three calibrations--the coefficient only, the coefficient and exponent, and the SAS® calibrations, to test the hypothesis that the relative frequency distributions resulting from each calibration was identical to the actual distribution. As with Military Ground, because the amount of data was small, the test was conducted using $\alpha = 0.10$, which resulted in a critical value of T (T_{crit}) = 8 . Therefore, if the calculated value of T (T_{calc}) proved to be less than or equal to 8, the hypothesis (H_0) that the relative frequency distributions of the two populations were identical could be rejected. Obviously, examining the results in Table 4-8, the hypothesis that the distributions were identical could be rejected for the coefficient only calibration. The hypothesis could not be rejected for the other predictions. These results appear logical when one recalls that the coefficient only calibration produced a model which actually predicted with a greater error than the uncalibrated model.

Table 4-8: Unmanned Space Wilcoxon Signed Rank Tests

	Pre-Calibration	Coeff Only	Coeff & Exp	SAS Log-Log
$T_{critical}$	8	8	8	8
$T_{calculated}$	17	1	15	10

Finally, the assumptions of linear regression were examined to determine if the assumptions of least squares best fit are met. To do this, once again, the Wilk-Shapiro/Rankit Plot of Residuals was used. As stated previously, if the standardized residuals are normally distributed, the plot of rankits against the ordered statistics should result in a straight line except for random variation. A systematic departure of the rankit plot from a linear trend indicates non-normality, as does the small value for the Wilk-Shapiro statistics. One or a few points departing from the linear trend near the extremes of the plot are indicative of outliers. As can be seen from the Unmanned Space Rankit Plot of Residuals , Figure 4-6, the residuals appear to have a heavy tail, indicating that the assumptions of linear regression are not met and the data for Unmanned Space are not a sample from a normal distribution. Examining additional results from the test for normality, included in Appendix D, page D-5, reinforces this finding. The second column, labeled *Prob* < *W*, contains the probability value which describes how doubtful the idea of normality is. Values close to zero indicate the data do not adhere to the assumptions of normality. The Unmanned Space statistics reveal a *W* value of 0.85725 and a *Prob* <*W* of 0.1392, thus supporting this researcher's initial findings.

```
                    Univariate Procedure
                     Variable=Residuals

                       Normal Probability Plot
 275+                                       *     ++++
     |                                        ++++
     |                                   +++++
     |                               +*++
  75+                           +++++
     |                      +*++*   *
     |                 *++*+
     |             ++++
-125+         *++++  *
     +----+----+----+----+----+----+----+----+----+----+
         -2        -1         0        +1        +2
```

Table 4-6: Unmanned Space Normality Test

Summary

This chapter has presented the results of the calibration effort on the Military Ground and Unmanned Space data sets. The techniques used and the resulting algorithms are presented with supporting documentation to be found in Appendices C and D.

V. Conclusions and Recommendations

Chapter Overview

This chapter addresses each of the issues identified for research in this effort and draws some conclusions based on the results of the REVIC calibration effort. Some recommendations for future areas of effort are offered.

Conclusions

ISSUE: *Input parameters which most strongly influence software costs* are not easily identified. The combined result of the 19 attributes which REVIC uses as a constant multiplier appear to be among the less influential of the many factors which determine software development cost and may, frequently, even result in a greater estimating error. For that reason, it is difficult to conclude that any one of the 19 parameters have a strong influence upon software development costs. The negligible effect of the attributes as constant multipliers were best demonstrated when comparisons were made between the results obtained using Boehm's coefficient and exponent calibration methodology and the results obtained using SAS®.

How can this be, when it is commonly acknowledged that software costs are influenced by such attributes as management abilities, support software tools, and personnel capabilities? One reason for such negligible effects may be due to the manner in which attribute data is made available. As a rule, the contractor provides the ratings for these parameters. As a result there remains the issue of standardizing the largely subjective opinions of various contractors to ensure that the ratings of nominal, high, etc. provide a standard and normalized measure. To do otherwise results in qualitative factors which are difficult to standardize across projects and contractors for calibration purposes. With this scenario, attributes may best be addressed by confining calibration to a single contractor and setting all nineteen attributes to "nominal" or to a value of "1." By doing this, one makes the assumption that, for an individual contractor, management skills, software tools available for development, and personnel capabilities and experience are relatively constant or fluctuate very little from project to project. Such an assumption can permit the software cost estimator to omit the subjective evaluation of the nineteen attributes and derive an algorithm based on only one independent variable, KDSI. Limiting data by contractor would also tend to minimize the impact of two factors which Walker felt contributed to a poor database--errors in data collection and lack of consistency among data sets. This is because we could expect the errors and omissions in data collection within a company to be more consistent than errors made across multiple companies and, therefore, it becomes unnecessary to quantify them as additional variables. This approach would also minimize, or tend to make constant such variables as: (1) observational bias, (2) inconsistent definitions, and (3) differences in local vs. global frames of reference--all problems which Boehm found to be frequent sources of software data collection problems.

ISSUE: The *effect of the software development environment on model performance* is also nebulous to this researcher, but appears to have a greater effect than the individual REVIC attributes. This was best illustrated when two environments, Unmanned Space and Military Ground in Support of Space, were incorrectly identified as belonging in the same environment. Even though the data appeared highly linear when plotted, further analysis revealed the sample data to be heteroscedastic and lacking a normal distribution. In contrast, the Military Ground data, when plotted, appeared to have a questionable linear relationship between KDSI and effort; however, tests for normality showed the sample data to consist of a normal distribution.

The probability that more than one independent variable (KDSI) may act as a major cost driver of software development cost is highly likely. However, no evidence of any prior research along this line was uncovered in the literature reviewed by this researcher during the literature search. A second qualitative or quantitative independent variable is suggested by the analysis of sample data for Military Ground (Figure 4-1, and 4-2). Upon examination of the sample data for possible cost driver candidates, variables which appeared as likely candidates included not only the contractor, but the program language , the software development model (e.g. waterfall, spiral, prototype or incremental), type of contract, and reliability requirements as evidenced by the level of documentation, quality assurance and testing required. As can be seen when comparing Military Ground and Unmanned Space, Military Ground data has a greater variance than does the Unmanned Space data. A major difference noted especially between the Military Ground and Unmanned Space data was the homogeneity of the program language and the number of contractors represented by the sample data used for calibration. The Unmanned Space sample data consisted almost entirely of projects developed in the Jovial language. Most data points used in calibrating to the Unmanned Space operating environment also come from one contractor. On the other hand, the Military Ground sample data represented six contractors and several program languages; and, as observed earlier, the Military Ground sample data also contained a greater variety of development methods and reliability requirements. When one compares the Unmanned Space scatter plot to the Military Ground scatter plot, an obvious difference is noted. What role the various variables identified above play in this difference is still undetermined. This researcher can only conclude, as Walker, Thibodeau, and others have, that model performance is very much environment dependent; and that we are, as of today, still unable to measure all major cost drivers of software development with any degree of objectivity. Therefore the problem remains one of identifying those environmental factors which are quantifiable and have the most impact on model performance.

ISSUE: *The calibration method which produced the best results* were the simultaneous coefficient and exponent calibration. There were two methods which accomplished this--Boehm's methodology using the Effort Adjustment Factor (EAF), and the simpler method of using a standard statistical software package and setting the EAF equal to one. Care should be used in the selection of a calibration method as some calibration efforts may result in a model which estimates less accurately than the default model. This was noted to be the case, in one instance, when the *coefficient only* calibration method was used during this research project.

ISSUE: Although calibration, in most instances, improved the estimating ability of REVIC, *the extent to which calibration influenced the accuracy of the software estimates* was most unimpressive. In no instance did calibration, using the single independent variable KDSI, produce a model that estimated within 25% of the actual value more than 50% of the time. This researcher has been led to conclude that calibration may only improve the accuracy of the REVIC software estimate in those cases where KDSI is the only variable and all other factors such as contractor, development model, software environment, and personnel and software attributes remain constant across projects.

ISSUE: *Circumstances in which REVIC may be most appropriate* are those circumstances where all independent variables except KDSI can be standardized and thus be excluded from the equation. Since the REVIC algorithm provided by the model can only recognize one independent variable, the REVIC model would not be an appropriate model to use for estimating when there seems to be a number of qualitative and/or quantitative cost drivers for software development cost. In situations where multiple independent variables are suspected, some other method of estimating software development cost should be used.

Recommendations

Several areas of further study need to be pursued. Included among these are:

(1) A search for other independent variables which drive software cost.

(2) A further examination of the results obtained when calibrations are limited to projects developed by a single contractor, making the calibration contractor specific.

(3) Further analysis of the impact the REVIC attributes have on effort and the accuracy of the values assigned to the ratings "nominal," "high", etc.

(4) Further analysis of the impact different software program languages have on estimating accuracy.

(5) The impact that different development methods (i.e. waterfall, prototype) have on effort and whether the development method is a cost driver.

(6) The kind of contract (FFP, CPAF, etc.) used in the development effort and the contract's effect on cost.

(7) Identification of other environmental factors which might impact estimating accuracy.

Summary

This chapter has summarized the insights and possibilities which have emerged as a result of the research effort. Some conclusions are reached and recommendations made for further study.

Appendix A. Glossary

Algorithm - A mathematical set of ordered steps leading to the optimal solution of a problem in a finite number of operations.

Analogy - an estimating methodology that compares the proposed system to similar, existing systems.

Attributes - Metrics used to measure some aspect of software development such as quality, complexity, or language and which serve as constant multipliers in the algorithms used in REVIC and COCOMO software cost estimating models.

Calibration - The adjustment of selected parameters of a given model to get an expected output with known inputs. In the world of statistics this effort is known as model building. For this research effort, the models already exits and will only be modified.

COCOMO - The **Con**structive **Co**st **Mo**del, a software cost estimating model developed by Barry Boehm.

Cost Estimating- The collecting and scientifically studying costs and related information on current and past activities as a basis for projecting costs as an input to the decision process for a future activity.

Cost Model - A tool consisting of one or more cost estimating relationships, estimating methodologies, or estimating techniques. Used to predict the cost of a system or some element of a system.

CSCI, CSC, and CSU - Large software development efforts are generally broken down into smaller, more manageable entities called computer software configuration items (CSCIs). Each CSCI may be further broken down into computer system components (CSCs) and each CSC may be further broken down into computer software units (CSUs).

Delivered Source Instructions - Equivalent to 1,000 source lines of code.

Embedded Programs - Software programs with tight constraints, such as on-board fighter aircraft programs.

Incremental Development - A software process model whose stages consist of expanding increments of an operational software product, with the direction of evolution being determined by operational experience.

Linear Cost Model - A cost estimating model which is linear in its parameters and in its independent variables. A linear cost model estimates costs using algorithms whose parameters or independent variables contain no exponents and are not multiplied or divided by another parameter or independent variable. A model which is linear in the parameters and the independent variable is also called a *first-order* model.

Macro Cost Estimation Model - A cost model that uses gross estimating parameters to arrive at an estimation.

Manmonth - Generally consists of 152 man hours of effort.

Normalization - The process of rendering constant or adjusting for known differences.

Organic Programs - Software programs which are usually small, stand-alone programs, such as payroll programs, developed by in-house teams.

Parameters - The parameters (B_0 and B_1) in a *linear cost model* are also called regression coefficients. B_1 is the slope of the regression line. B_0 is the Y intercept of the regression line. Parameters of a *normal distribution* are the mean (μ) and the standard deviation (σ).

Parametric Model - A model that uses one or more cost estimating relationships or algorithms, based on the project's technical, lphysica, other characteristic, to estimate costs associated with the development of that item.

Program Evaluation and Review Technique - A network or diagram consisting of arrows and end points. The network represents project activities, their associated durations, and precedence relationships between pairs of activities.

Phase Sensitivity - a procedure which examines the various phases in software development to determine the impact of changing specific conditions in a particular phase will have upon the variation of the estimate.

Prototype Development Method - An *iterative* software process model.

Rayleigh Distribution - A probability distribution whose curve is characterized by a rather steep buildup as coding begins, followed by a long tapering-off period before the system is ready for delivery. It can also be used to describe the rate of defect discovery and the application of people to a project.

Regression Analysis - A statistical tool that uses the relation between two or more quantitative variables so that one variable can be predicted from the other, or others.

REVIC - A software cost estimating model developed by Raymond Kile.

Semi-detached Programs - Programs containing both embedded and organic characteristics, such as flight simulator programs.

Sensitivity Analysis - A procedure that examines the variation in an estimate subject to changing specific conditions on which it was based.

Software - The combination of computer programs, data, and documentation which enables computer equipment to perform computational or central functions.

Software Maintenance - Since software does not wear out, SW maintenance refers to corrective, adaptive, or perfective changes made to software.

Software Development Cycle - The software development cycle is typically broken into 8 phases: (1) System Requirements Analysis and Design, (2) Software Requirements Analysis, (3) Preliminary Design, (4) Detailed Design, (5) Code and CSU Testing, (6) CSC Integration and Testing, (7) CSCI Testing, and (8) System Testing.

Source Lines of Code (SLOC) - All program instructions created by the project personnel and processed into machine code. It includes job control, format statements, etc., but does not include comment statements and unmodified utility software.

Spiral Software Development Model - A *risk driven*, cyclical software process model with a repeating set of activities performed on an increasingly more detailed product. It can accommodate most other process models, such as the Waterfall Development Model. In addition, it provides guidance as to which combination of other models best fits a given software situation.

Validation - Testing a specific model using known inputs and establishing the output to within some error range. This is independent and non-iterative with calibration. In the world of statistics, this is often called cross-validation since it will use a portion of an original data set kept out of the model building/calibration effort.

Waterfall Development Model - A *document driven* software process model which stipulates that software be developed in successive stages. It determines the order of the stages involved in software development and evolution and establishes the transition criteria for progressing from one stage to the next.

Appendix B. Acronyms

ACAP -- Analyst's Capability
ADSI -- Adapted Delivered Source Instructions
AEXP -- Applications Experience
AFCAA -- Air Force Cost Analysis Agency
CER -- Cost Estimating Relationship
CIM -- Corporate Information Management
CM -- Code Modification
COCOMO -- Constructive Cost Model
COSTMODL -- Cost Model
CPLX -- Code Complexity
CSC -- Computer Software Component
CSCI -- Computer Software Configuration Item
CSU -- Computer Software Unit
DATA -- Data Base Size
DM -- Design Modification
DoD -- Department of Defense
DSI -- Delivered Source Instructions
EAF -- Effort Adjustment Factor also denoted as Π
EDSI -- Equivalent Delivered Source Instructions
HOL -- Higher Order Language
HPCC -- High Performance Computing and Communications
IM -- Retesting of Modified code
KDSI -- Thousands of Delivered Source Instructions
LEXP -- Language Experience
MCR -- Management Consulting and Research, Inc.
MM -- Man-month
MMRE -- Mean Mangitude of Relative Error
MODP -- Modern Programming Practices
MRE -- Magnitude of Relative Error
PCAP -- Programmer's Capability
PERT -- Program Evaluation and Review Technique
PRED -- Prediction Level Test
PRICE-S -- Programmed Review of Information for Costing and Evaluation Software
RELY -- Required Reliability
REVIC -- Revised Enhanced Version of Intermediate COCOMO
RISK -- Risk associated with platform
RMS -- Root Mean Square Error
RRMS -- Relative Root Mean Square Error
RUSE -- Required Reusability

RVOL -- Requirements Volatility
SAS® -- System for Elementary Statistical Analysis by SAS Institute, Inc.
SASET --Software Architecture, Sizing, and Estimating Tool
SBA -- Standards-Based Architecture
SCED -- Schedule Compression/Stretch Out
SDC -- Systems Development Corporation
SECU -- Security Classification
SEER-SEM -- System Estimation & Evaluation of Resources Software Estimation Model
SLOC -- Source Lines of Code
SMC -- Space and Missile Systems Center
SSCAG -- Space Systems Cost Analysis Group
SWDB -- Software Database
TIME -- Processing or Throughput Time Constraints
TOOL -- Design and Programming Tools
TURN -- Turnaround Time
VEXP -- Virtual Machine Experience
VIRT -- Virtual Machine Volatility

Appendix C. Military Ground Worksheets

Military Ground Operating Environment

SMC SWDB PARAMETER	REVIC Equiv	Project No. 2497	2501	2510	2517	2521	2526	2527	2528	2610	2611	2612
4.3.01 Appl Cmplx	None											
4.3.02 Turn-around	None											
4.3.03 Reqr Volatil	**RVOL**	NOM	VH	HI	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM
4.3.04 Rehost Reqr	None											
4.3.05 Display Reqr	None											
4.3.06 Reuse Reqr	**RUSE**	NOM	HI	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM
4.3.07 Security Level	**SECU**	(Note 1)										
4.3.08 Memory Constr	**STOR**	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM
4.3.09 Time Constr	**TIME**	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM
4.3.10 Real Time	None											
4.8.01 Pers Exp	**AEXP**	LO	NOM	HI	HI	NOM	NOM	NOM	NOM	NOM	NOM	NOM
4.8.02 Pers Cap	**P/ACAP**	NOM	HI	NOM	HI	NOM	NOM	NOM	NOM	NOM	NOM	*NOM
4.8.03 Target Virt	None											
4.8.04 Host Virt	None											
4.8.05 Prog Lang	**LEXP**	NOM	HI	HI	VH	LO	LO	LO	LO	LO	LO	LO
4.8.06 Dev Exp	None											
4.8.07 Dev Sys Exp	**VEXP**	HI	VH	NOM	VH	HI	HI	HI	HI	HI	HI	HI
4.8.08 Target Sys Exp	**VEXP**	HI	VH	NOM	VH	HI	HI	HI	HI	HI	HI	HI
4.23.01 Inher Dif	**CPLX**	HI	VH	HI	HI	VL	HI	HI	HI	HI	HI	HI
4.23.02 Turn Time	**TURN**	HI	LO	(LO)	LO	LO	LO	LO	LO	LO	LO	LO
4.23.03 Term Respon	None											
4.23.04 Dev Sys Vol	**VIRT**	HI	LO	LO	NOM	NOM	LO	LO	LO	LO	LO	LO
4.23.05 Spec Level	**RELY**	VH	XH	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM
4.23.06 QA Level	**RELY**	NOM	VH	NOM	LO	NOM	LO	LO	LO	LO	LO	LO
4.23.07 Test Level	**RELY**	(Note 2)										
4.23.08 Mult Level	None											
4.23.09 Resour Dedic	None											
4.23.10 Res/ Spprt	None											
4.23.11 No. Shifts	None											
4.23.12 Amt Travel	None											
4.23.13 Modrn Pract	**MODP**	VH	HI	NOM	LO	HI	NOM	NOM	NOM	NOM	NOM	NOM
4.23.14 Auto Tool	**TOOL**	VH	LO	NOM	LO	HI	NOM	NOM	NOM	NOM	NOM	NOM

Note 1: Attribute not used due to incomplete data in records.
Note 2: Attribute not used due to incomplete data in records.
Note 3: Attribute denoted by (*) are subjective opinions and were not available from the record.

Military Ground

		REVIC DATA POINTS									
Proj. No.	**2497**	**2501**	**2510**	**2517**	**2521**	**2526**	**2527**	**2528**	**2610**	**2611**	**2612**
Dev Yr.	1993	1993	1993	1992	1992	1992	1992	1992	1992	1992	1992
Language	Ada	Ada	C	Assy	Cobol	Cobol	Cobol	Cobol	Cobol	Cobol	Cobol
Dev Model	Mod WF	Mod WF	Unknwn	Unknwn	Incrmntl	Prototyp	Prototyp	Prototyp	Prototyp	Prototyp	Prototyp
Type Contract	FFP	FFP	CPAF	Unknwn	FFP	FFP	FFP	FFP	FFP	FFP	FFP
Mos. in Dev	40	21	24	48	45	57	57	57	57	57	57
DSI-Actual	10,000	106,200	43,437	90,000	97,087	6,681	7,457	21,588	14,536	11,840	9,899
New	10,000	45,000	43,437	76,200	97,087	6,681	7,457	21,588	14,536	11,840	9,899
EDSI*		61,200		13,800							
Reused		120,000		13,800							
% DM		30		100							
% CM		30		100							
% IM		100		100							
EFFORT											
Actual	80.0	418.0	181.2	196.0	735.0	202.0	225.0	652.0	439.0	358.0	299.0
Normalizd											
152 hr/mm	84.2	475.8	182.4	206.3	836.5	202.0	225.0	652.0	439.0	358.0	299.0
REVIC Equiv	89.4	542.6	193.6	235.3	954.0	208.8	232.5	673.8	453.7	370.0	309.0
REVIC Est.											
Pre Calibrtn	66.2	1,586.1	306.7	684.4	522.5	27.0	30.6	109.7	68.3	53.4	43.1
Post Calibrtn											
Coeff only	47.4	1135.0	219.5	536.4	373.9	19.2	22.0	78.5	48.9	38.2	30.8
Coeff & Exp	315.9	1348.7	501.0	656.7	474.4	172.0	181.1	298.4	247.9	225.0	206.9
SAS (log)	224.9	639.6	430.7	796.7	614.7	188.0	197.5	316.1	338.0	242.4	285.1
REVIC EAF	1.126	1.324	0.895	0.697	0.486	0.838	0.838	0.838	0.838	0.838	0.838
Phase Incl											
SW Req	x	x	x	x	x						
Prelim Dsn	x	x	x	x	x						
Detail Dsn	x	x	x	x	x	x	x	x	x	x	x
C&U Test	x	x	x	x	x	x	x	x	x	x	x
CSC T&I	x	x	x	x	x	x	x	x	x	x	x
CSCI Test	x	x	x	x	x	x	x	x	x	x	x
Sys T&I		x		x	x	x	x	x	x	x	x
OT&E		x		x	x						

*EDSI = Equivalent DSI = (ADSI) X (AAF/100), where
ADSI = Adapted DSI or SLOC, and
AAF = Adaptation Adjustment Factor = .40 (DM) + .30 (CM) + .30 (IM).

Military Ground - Coefficient only calibration							
Proj No.	KDSI	EAF	MMest	MMi	Qi	MMiQ	Q*Q
2497	10	1.126	66.2	89.4	17.85	1,595.42	318.48
2501	106.2	1.324	1586.1	542.6	357.47	193,962.24	127,783.51
2510	43.437	0.895	306.7	193.6	82.65	16,001.48	6,831.40
2517	Control						
2521	97.087	0.486	522.5	954.0	117.82	112,403.01	13,882.23
2526	6.681	0.838	27.0	208.8	8.19	1,709.17	67.01
2527	7.457	0.838	30.6	232.5	9.34	2,171.42	87.23
2528	21.588	0.838	109.7	673.8	33.44	22,533.57	1,118.40
2610	Control						
2611	11.84	0.838	53.4	370.0	16.27	6,018.22	264.56
2612	Control						
			Totals			356,394.52	150,352.81
C(mean) =	***Coefficient***	***2.370***					
MM=2.370 * (KDSI)^1.20 * (EAF)							
Military Ground - Coefficient and Exponent Calibration							
							log(MM/EAF)*
Proj No.	KDSI	EAF	MM	log KDSI	log (KDSI)^2	log(MM/EAF)	* log(KDSI)
				(a1)	(a2)	(d0)	(d1)
2497	10	1.126	89.4	1.000	1	1.900	1.900
2501	106.2	1.324	542.6	2.026	4.105	2.613	5.293
2510	43.437	0.895	193.6	1.638	2.683	2.335	3.825
2517	Control						
2521	97.087	0.486	954.0	1.987	3.948	3.293	6.543
2526	6.681	0.838	208.8	0.825	0.681	2.396	1.977
2527	7.457	0.838	232.5	0.873	0.762	2.443	2.133
2528	21.588	0.838	673.8	1.334	1.780	2.905	3.876
2610	Control						
2611	11.84	0.838	370.0	1.073	1.151	2.645	2.838
2612	Control						
			Totals	10.756	16.11	20.53	28.38
log c(mean)=log coefficient			1.928745				
c(mean) =	***84.868***						
b(mean) =	***0.474***						
Therefore:							
MM=84.868 (KDSI)^0.474 * EAF							

MILITARY GROUND

Before Calibration								
Proj No.	Y(act)	Y(pred)	act-mean	(act-m)sq	pred-mean	(prd-m)sq	(act-predt)	(act-prd)sq
2497	89.4	66.2						
2501	542.6	1,586.1						
2510	193.6	306.7						
2517	235.3	684.4	-172.79	29,855.52	276.3125	76,348.60	-449.1	201,690.81
2521	954.0	522.5						
2526	208.8	27.0						
2527	232.5	30.6						
2528	673.8	109.7						
2610	453.7	68.3	45.61	2,080.50	-339.788	115,455.55	385.4	148,533.16
2611	370.0	53.4						
2612	309.0	43.1	-99.09	9,818.33	-364.988	133,215.88	265.9	70,702.81
Sum	3,264.7		(226.3)	41,754.4	(428.5)	325,020.0	202.2	420,926.8
Mean	408.09							

After calibration of coefficient								
Proj No.	Y(act)	Y(pred)	act-mean	(act-m)sq	pred-mean	(prd-m)sq	(act-predt)	(act-prd)sq
2497	89.4	47.40						
2501	542.6	1,135.00						
2510	193.6	219.50						
2517	235.3	536.40	-172.79	29,855.52	128.3125	16,464.10	-301.1	90,661.21
2521	954.0	373.90						
2526	208.8	19.20						
2527	232.5	22.00						
2528	673.8	78.50						
2610	453.7	48.90	45.61	2,080.50	-359.188	129,015.66	404.8	163,863.04
2611	370.0	38.20						
2612	309.0	30.80	-99.09	9,818.33	-377.288	142,345.86	278.2	77,395.24
Sum	3,264.70		(226.26)	41,754.35	(608.16)	287,825.62	381.90	331,919.49
Mean	408.09							

After calibration of coefficient and exponent								
Proj No.	Y(act)	Y(pred)	act-mean	(act-m)sq	pred-mean	(prd-m)sq	(act-predt)	(act-prd)sq
2497	89.4	315.90						
2501	542.6	1,348.70						
2510	193.6	501.00						
2517	235.3	656.70	-172.79	29,855.52	248.6125	61,808.18	-421.4	177,577.96
2521	954.0	474.40						
2526	208.8	172.00						
2527	232.5	181.10						
2528	673.8	298.40						
2610	453.7	247.90	45.61	2,080.50	-160.188	25,660.04	205.8	42,353.64
2611	370.0	225.00						
2612	309.0	206.90	-99.09	9,818.33	-201.188	40,476.41	102.1	10,424.41
Sum	3,264.70		(226.26)	41,754.35	(112.76)	127,944.62	(113.50)	230,356.01
Mean	408.09							

After calibration using SAS								
Proj No.	Y(act)	Y(pred)	act-mean	(act-m)sq	pred-mean	(prd-m)sq	(act-predt)	(act-prd)sq
2497	89.4	224.90						
2501	542.6	639.64						
2510	193.6	430.69						
2517	235.3	796.70	-172.79	29,855.52	388.6125	151,019.68	-561.4	315,169.96
2521	954.0	614.74						
2526	208.8	188.00						
2527	232.5	197.53						
2528	673.8	316.10						
2610	453.7	338.00	45.61	2,080.50	-70.0875	4,912.26	115.7	13,386.49
2611	370.0	242.35						
2612	309.0	285.10	-99.09	9,818.33	-122.988	15,125.93	23.9	571.21
Sum	3,264.70		(226.26)	41,754.35	195.54	171,057.86	(421.80)	329,127.66
Mean	408.09							

Military Ground Normality Test

Univariate Procedure

```
Variable=Residuals

                              Moments

           N                   8  Sum Wgts           8
           Mean          51.3375  Sum            410.7
           Std Dev      215.1114  Variance    46272.91
           Skewness     0.368567  Kurtosis    -1.00182
           USS          344994.7  CSS         323910.4
           CV           419.0142  Std Mean    76.05336
           T:Mean=0     0.675019  Pr>|T|        0.5213
           Num ^= 0            8  Num > 0            5
           M(Sign)             1  Pr>=|M|       0.7266
           Sgn Rank            4  Pr>=|S|       0.6406
           W:Normal     0.935542  Pr<W          0.5722


                          Quantiles(Def=5)

            100% Max       357.7          99%       357.7
             75% Q3       233.45          95%       357.7
             50% Med       27.85          90%       357.7
             25% Q1      -116.25          10%      -237.1
              0% Min      -237.1           5%      -237.1
                                           1%      -237.1
            Range          594.8
            Q3-Q1          349.7
            Mode          -237.1


                              Extremes

               Lowest    Obs      Highest    Obs
              -237.1(      3)       20.7(      5)
              -135.5(      1)         35(      6)
                 -97(      2)      127.6(      8)
                20.7(      5)      339.3(      4)
                  35(      6)      357.7(      7)


        Stem Leaf                     #             Boxplot
           3 46                       2                |
           2                                       +------+
           1 3                        1            |      |
           0 24                       2            *--+--*
          -0                                       |      |
          -1 40                       2            +------+
          -2 4                        1                |
             ----+----+----+----+
         Multiply Stem.Leaf by 10**+2
```

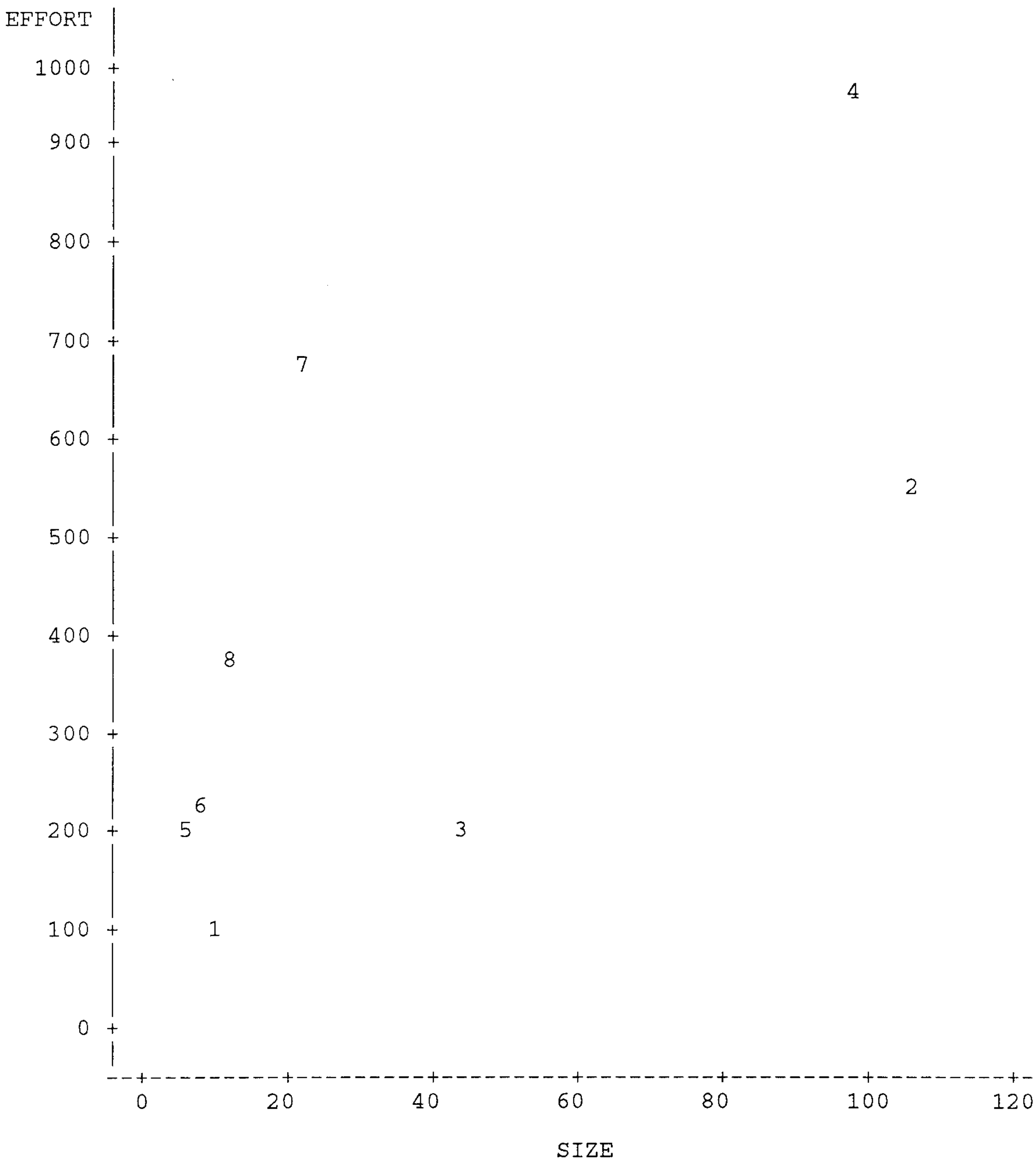
```
CSC>ed milgrnd.dat
89.4 10
542.6 106.2
193.6 43.437
954.0 97.087
208.8 6.681
232.5 7.457
673.8 21.588
370.0 11.840
```

Military Ground

OBS	EFFORT	SIZE	LEFFORT	LSIZE
1	89.4	10.000	4.49312	2.30259
2	542.6	106.200	6.29637	4.66532
3	193.6	43.437	5.26579	3.77131
4	954.0	97.087	6.86066	4.57561
5	208.8	6.681	5.34138	1.89927
6	232.5	7.457	5.44889	2.00915
7	673.8	21.588	6.51293	3.07214
8	370.0	11.840	5.91350	2.47148

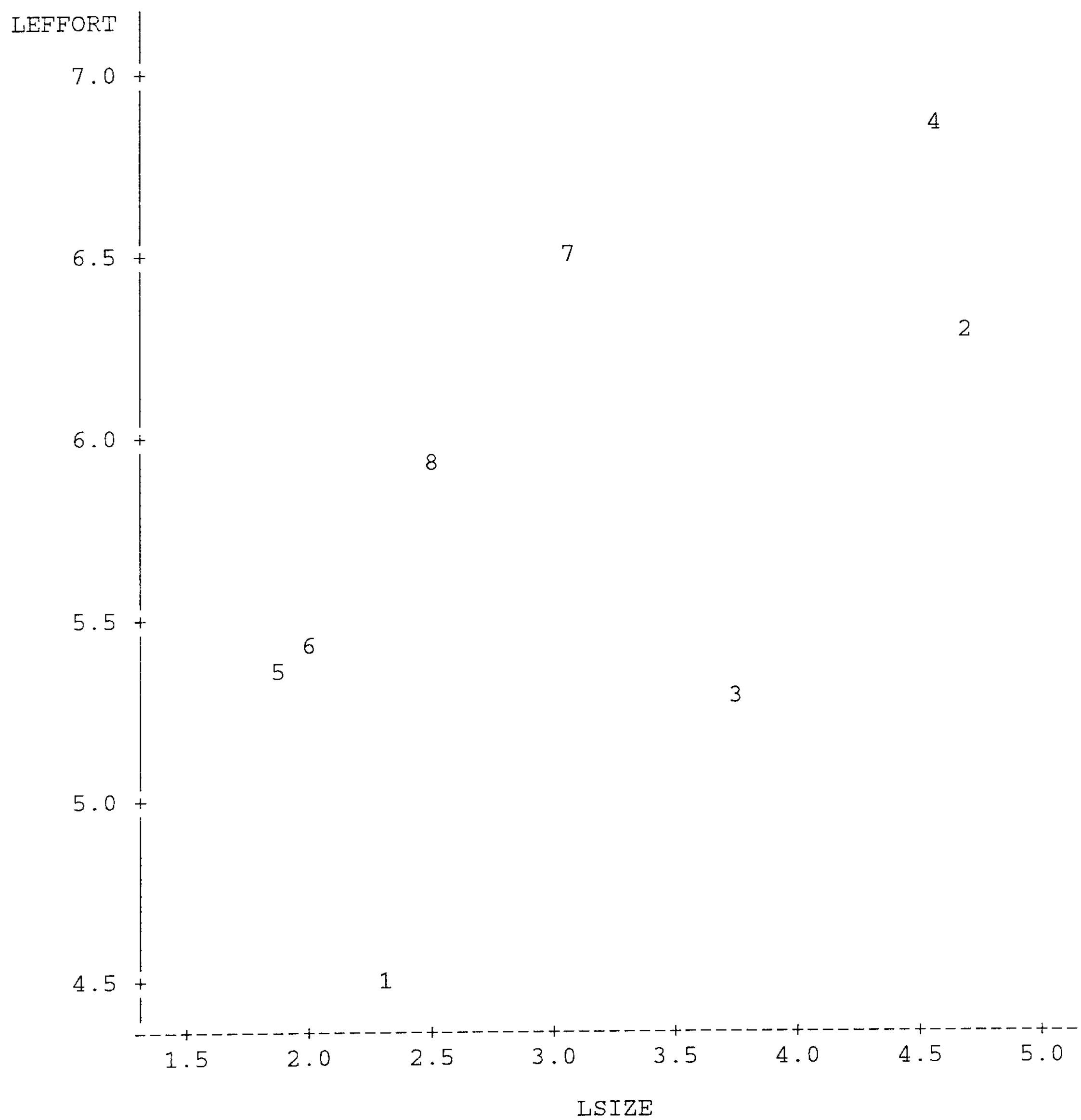
Military Ground

Plot of EFFORT*SIZE. Legend: 1= 1st obs, 2= 2nd obs, etc.



Military Ground

Plot of LEFFORT*LSIZE. Legend: 1 = 1st obs, 2 = 2nd obs, etc.



Military Ground

Model: MODEL1
Dependent Variable: EFFORT

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	278924.90232	278924.90232	5.112	0.0645
Error	6	327356.04643	54559.34107		
C Total	7	606280.94875			

Root MSE	233.57941	R-square	0.4601
Dep Mean	408.08750	Adj R-sq	0.3701
C.V.	57.23758		

Parameter Estimates

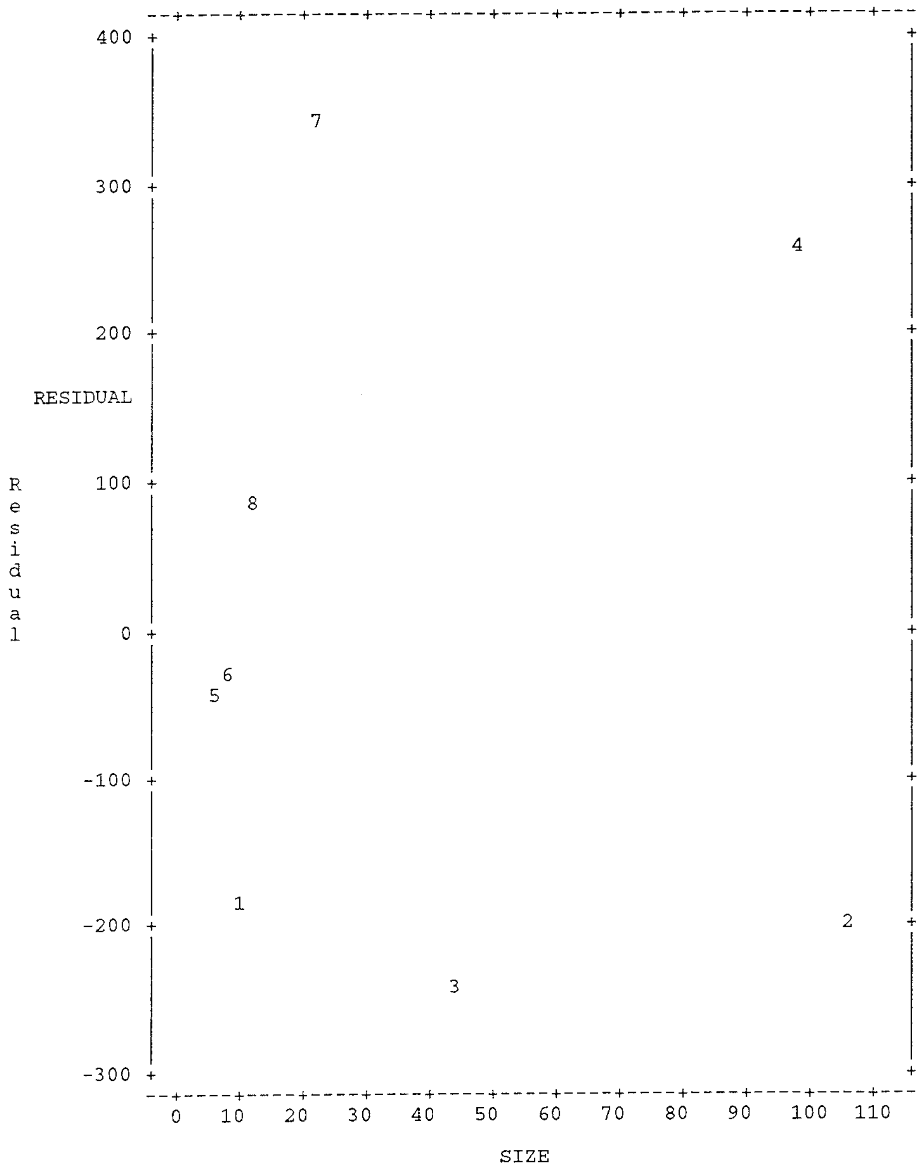
Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	223.344505	116.17202787	1.923	0.1029
SIZE	1	4.857024	2.14813317	2.261	0.0645

Obs	Dep Var EFFORT	Predict Value	Std Err Predict	Lower95% Mean	Upper95% Mean	Lower95% Predict
1	89.4000	271.9	102.211	21.8140	522.0	-352.0
2	542.6	739.2	168.108	327.8	1150.5	34.9790
3	193.6	434.3	83.394	230.3	638.4	-172.6
4	954.0	694.9	151.362	324.5	1065.3	13.8393
5	208.8	255.8	106.568	-4.9672	516.6	-372.4
6	232.5	259.6	105.522	1.3606	517.8	-367.6
7	673.8	328.2	89.824	108.4	548.0	-284.2
8	370.0	280.9	99.933	36.3248	525.4	-340.8

Obs	Upper95% Predict	Residual
1	895.8	-182.5
2	1443.3	-196.6
3	1041.2	-240.7
4	1376.0	259.1
5	884.0	-46.9943
6	886.7	-27.0633
7	940.6	345.6
8	902.5	89.1483

Sum of Residuals	0
Sum of Squared Residuals	327356.0464
Predicted Resid SS (Press)	673944.0684

Military Ground



Military Ground

Model: MODEL2
Dependent Variable: LEFFORT

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	1.71064	1.71064	4.106	0.0891
Error	6	2.49990	0.41665		
C Total	7	4.21054			

Root MSE	0.64548	R-square	0.4063
Dep Mean	5.76658	Adj R-sq	0.3073
C.V.	11.19354		

Parameter Estimates

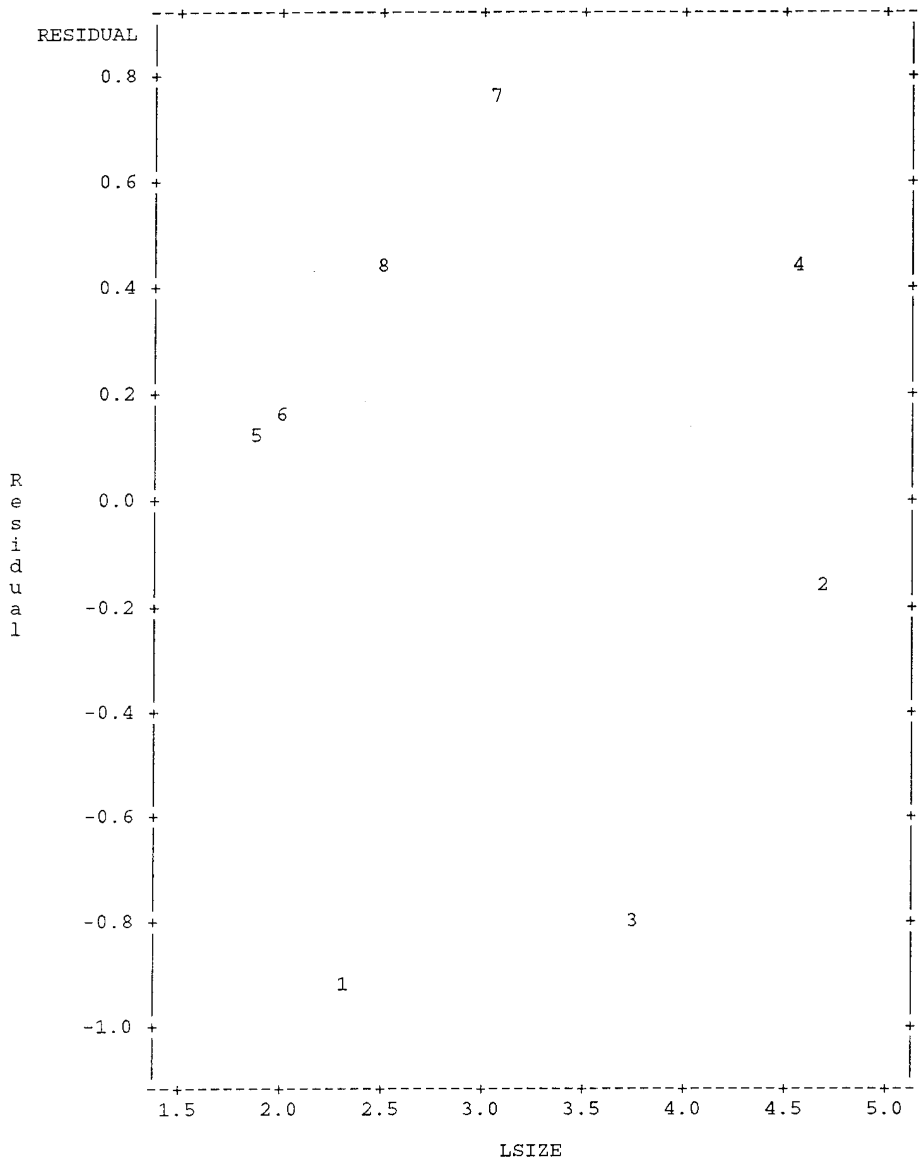
Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	4.397071	0.71337287	6.164	0.0008
LSIZE	1	0.442369	0.21831884	2.026	0.0891

Obs	Dep Var LEFFORT	Predict Value	Std Err Predict	Lower95% Mean	Upper95% Mean	Lower95% Predict
1	4.4931	5.4157	0.286	4.7147	6.1167	3.6876
2	6.2964	6.4609	0.412	5.4535	7.4682	4.5875
3	5.2658	6.0654	0.272	5.4005	6.7302	4.3517
4	6.8607	6.4212	0.396	5.4533	7.3890	4.5688
5	5.3414	5.2372	0.347	4.3885	6.0860	3.4442
6	5.4489	5.2859	0.329	4.4804	6.0914	3.5129
7	6.5129	5.7561	0.228	5.1975	6.3146	4.0808
8	5.9135	5.4904	0.266	4.8399	6.1408	3.7822

Obs	Upper95% Predict	Residual
1	7.1437	-0.9225
2	8.3342	-0.1645
3	7.7791	-0.7996
4	8.2736	0.4395
5	7.0303	0.1041
6	7.0588	0.1630
7	7.4314	0.7568
8	7.1985	0.4231

Sum of Residuals	0
Sum of Squared Residuals	2.4999
Predicted Resid SS (Press)	3.9142

Military Ground



Appendix D. Unmanned Space Worksheets

Unmanned Space Operating Environment

SMC SWDB PARAMETER	REVIC Equiv	Project No. 74	75	76	77	78	79	80	81	82	83	306	2516	2518
4.3.01 Appl Cmplx	None													
4.3.02 Turnaround	None													
4.3.03 Reqr Volati	RVOL	NOM	VH	VH	VH	NOM	NOM	NOM	NOM	NOM	NOM	LO	HI	NOM
4.3.04 Rehost Requ	None													
4.3.05 Display Req	None													
4.3.06 Reuse Requi	**RUSE**	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM
4.3.07 Security Le	**SECU**	(Note 1)												
4.3.08 Memory Cons	**STOR**	(Note 1)												
4.3.09 Time Constr	**TIME**	(Note 1)												
4.3.10 Real Time	None													
4.8.01 Pers Exp	**AEXP**	(Note 1)												
4.8.02 Pers Cap	**P/ACAP**	(Note 1)												
4.8.03 Target Virt	None													
4.8.04 Host Virt	None													
4.8.05 Prog Lang	**LEXP**	LO	LO	NOM	LO	LO	LO	LO	LO	LO	LO	NOM	HI	HI
4.8.06 Dev Meth Ex	None													
4.8.07 Dev Sys Exp	**VEXP**	LO	LO	NOM	LO	LO	LO	LO	LO	LO	LO	LO	NOM	NOM
4.8.08 Target Sys	**VEXP**	LO	LO	NOM	LO	LO	LO	LO	LO	LO	LO	VL	NOM	NOM
4.23.01 Inher Dif	**CPLX**	HI	VH	NOM	VH	NOM	NOM	NOM	NOM	NOM	LO	LO	NOM	HI
4.23.02 Turn Time	**TURN**	HI	HI	HI	HI	HI	HI	HI	HI	HI	HI	VL	VL	LO
4.23.03 Term Respo	None													
4.23.04 Dev Sys Vo	**VIRT**	HI	HI	HI	HI	HI	HI	HI	HI	HI	HI	LO	LO	LO
4.23.05 Spec Level	**RELY**	HI	HI	HI	HI	HI	HI	HI	HI	HI	HI	LO	LO	LO
4.23.06 QA Level	**RELY**	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	VL	VL	LO
4.23.07 Test Level	**RELY**	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	LO	HI
4.23.08 Mult Site	None													
4.23.09 Resour Ded	None													
4.23.10 Res/Supprt	None													
4.23.11 No. Shifts	None													
4.23.12 Amt Travel	None													
4.23.13 Modrn Prac	**MODP**	VH	VH	VH	VH	VH	VH	VH	VH	VH	VH	NOM	NOM	LO
4.23.14 Auto Tool	**TOOL**	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	NOM	LO	NOM	LO

Note 1: Attribute not used due to incomplete data in records.
This is equivalent to setting the attribute at a value of "nominal" or "1".

		REVIC DATA POINTS											
Proj. No.	**74**	**75**	**76**	**77**	**78**	**79**	**80**	**81**	**82**	**83**	**306**	**2561**	**2518**
Dev Yr.	1985	1985	1985	1985	1986	1985	1985	1985	1986	1985	1988	1989	1988
Dev Model	WF	WF	WF	WF	WF	WF	WF	WF	WF	WF	Incr-4	Mod WF	Unknown
Type Contract	FFP	FFP	FFP	FFP	FFP	FFP	FFP	FFP	FFP	FFP	CPFF	Unknown	Unknown
Mos. in Dev													
DSI-Actual	11,700	116,800	14,000	56,200	48,300	50,300	69,450	22,900	16,300	6,800	9,400	48,814	18,004
New	11,700	116,800	14,000	56,200	48,300	50,300	69,450	22,900	16,300	6,800	9,000	48,675	17,344
EDSI*											400	138.8	660
Reused											500	400	660
% DM											50	8	100
% CM											100	5	100
% IM											100	100	100
EFFORT													
Actual	80.0	912.0	115.0	523.0	478.0	432.0	296.0	164.0	140.0	57.0	90.0	117	96
Normalizd													
152 hr/mm	80.0	912.0	115.0	523.0	478.0	432.0	296.0	164.0	140.0	57.0	90.0	123.2	101.1
REVIC Equiv	80.0	912.0	115.0	523.0	478.0	432.0	296.0	164.0	140.0	57.0	69.4	197.4	115.3
REVIC Est.													
Pre Calibrtn	86.6	2,135.6	109.5	887.7	412.5	433.1	637.9	168.5	112.1	33.4	20.4	269.5	142.5
Post Calibrtn													
Coeff only	40.0	984.9	50.5	409.4	190.4	199.6	294.2	77.7	51.7	15.6	9.4	124.2	65.8
Coeff & Exp	106.0	1,041.9	124.7	580.4	286.6	296.0	383.1	157.7	120.1	50.8	27.3	166.6	146.9
SAS (log)	85.6	657.3	100.4	343.8	300.7	311.6	414.7	155.2	114.85	52.9	70.5	303.5	125.4
REVIC EAF	1.366	2.131	1.392	2.131	1.188	1.188	1.188	1.188	1.188	1.009	0.566	0.684	1.001
Phase Incl													
SW Req												x	x
Prelim Dsn	x	x	x	x	x	x	x	x	x	x	x	x	x
Detail Dsn	x	x	x	x	x	x	x	x	x	x	x	x	x
C&U Test	x	x	x	x	x	x	x	x	x	x	x	x	x
CSC T&I	x	x	x	x	x	x	x	x	x	x	x		x
CSCI Test	x	x	x	x	x	x	x	x	x	x		x	x
Sys T&I													x
OT&E													x

*EDSI = Equivalent DSI = (ADSI) X (AAF/100), where
ADSI = Adapted DSI or SLOC, and
AAF = Adaptation Adjustment Factor = .40 (DM) + .30 (CM) + .30 (IM).

Unmanned Space - Coefficient only calibration							
Proj No.	KDSI	EAF	MMest	MMi	Qi	MMiQ	Q*Q
74	11.700	1.366	86.6	80.00	26.14	2,091.05	683.20
75	116.800	2.131	2,135.6	912.00	644.93	588,179.25	415,939.08
76	14.000	1.392	109.5	115.00	33.04	3,799.19	1,091.40
77	CONTROL					0.00	0.00
78	CONTROL					0.00	0.00
79	50.300	1.188	433.1	432.00	130.83	56,517.35	17,115.75
80	69.450	1.188	637.9	296.00	192.67	57,031.51	37,123.27
81	22.900	1.188	168.5	164.00	50.89	8,345.70	2,589.63
82	CONTROL					0.00	0.00
83	6.800	1.009	33.4	57.00	10.07	573.82	101.35
306	CONTROL			58.00		0.00	0.00
2516	48.814	0.684	269.5	197.46	72.66	14,347.91	5,279.82
2518	18.004	1.001	142.5	115.24	32.13	3,702.37	1,032.18
			Totals			734,588.14	480,955.68
C(mean) =	*Coefficient*	***1.5274***					
MM=1.527 * (KDSI)^1.20 * (EAF)							

Unmanned Space - Coefficient and Exponent Calibration							
							log(MM/EAF)*
Proj No.	KDSI	EAF	MMi	log KDSI	log (KDSI)^2	log(MM/EAF)	log(KDSI)
				(a1)	(a2)	(d0)	(d1)
74	11.700	1.366	80.00	1.068	1.141	1.768	1.888
75	116.800	2.131	912.00	2.067	4.272	2.631	5.439
76	14.000	1.392	115.00	1.146	1.313	1.917	2.197
77	CONTROL						
78	CONTROL						
79	50.300	1.188	432.00	1.702	2.897	2.561	4.358
80	69.450	1.188	296.00	1.842	3.393	2.396	4.414
81	22.900	1.188	164.00	1.360	1.850	2.140	2.910
82	CONTROL						
83	6.800	1.009	57.00	0.833	0.694	1.752	1.459
306	CONTROL		58.00				
2516	48.814	0.684	197.46	1.689	2.853	2.460	4.155
2518	18.004	1.001	115.24	1.255	1.575	2.061	2.587
			Totals	12.962	19.987	19.687	29.409
log c(mean)=log coefficient			1.035407				
c(mean) =	***10.8494***						
b(mean) =	***0.800***						
Therefore:							
MM=10.8489 (KDSI)^0.8 * EAF							

Unmanned Space Ope								
Before Calibration								
Proj No.	Y(act)	Y(pred)	act-mean	(act-m)sq	pred-mean	(prd-m)sq	(act-predt)	(act-prd)sq
74	80.0	86.6						
75	912.0	2,135.6						
76	115.0	109.5						
77	523.0	887.7	259.81	67,501.81	624.51	390,014.13	-364.7	133,006.09
78	478.0	412.5	214.81	46,143.81	149.31	22,293.81	65.5	4,290.25
79	432.0	433.1						
80	296.0	637.9						
81	164.0	168.5						
82	140.0	112.1	-123.19	15,175.50	(151.09)	22,827.85	27.9	778.41
83	57.0	33.4						
306	69.4	20.4	-193.79	37,554.13	(242.79)	58,946.44	49	2,401.00
2516	197.4	269.5						
2518	115.3	142.5						
Sum	2,368.7	1,432.7	157.6	166,375.3	379.9	494,082.2	(222.3)	140,475.8
Mean	263.19							
After calibration of coefficient								
Proj No.	Y(act)	Y(pred)	act-mean	(act-m)sq	pred-mean	(prd-m)sq	(act-predt)	(act-prd)sq
74	80.0	40.00						
75	912.0	984.90						
76	115.0	50.50						
77	523.0	409.40	259.81	67,501.81	146.2111	21,377.69	113.6	12,904.96
78	478.0	190.40	214.81	46,143.81	-72.7889	5,298.22	287.6	82,713.76
79	432.0	199.60						
80	296.0	294.20						
81	164.0	77.70						
82	140.0	51.70	-123.19	15,175.50	-211.489	44,727.55	88.3	7,796.89
83	57.0	15.60						
306	69.4	9.40	-193.79	37,554.13	-253.789	64,408.80	60	3,600.00
2516	197.4	124.20						
2518	115.3	65.80						
Sum	2,368.7	660.9	157.6	166,375.3	(391.9)	135,812.3	549.5	107,015.6
Mean	263.19							
After calibration of coefficient and exponent								
Proj No.	Y(act)	Y(pred)	act-mean	(act-m)sq	pred-mean	(prd-m)sq	(act-predt)	(act-prd)sq
74	80.0	106.00						
75	912.0	1,041.90						
76	115.0	124.70						
77	523.0	580.40	259.81	67,501.81	317.2111	100,622.89	-57.4	3,294.76
78	478.0	286.60	214.81	46,143.81	23.41111	548.08	191.4	36,633.96
79	432.0	296.00						
80	296.0	383.10						
81	164.0	157.70						
82	140.0	120.10	-123.19	15,175.50	-143.089	20,474.43	19.9	396.01
83	57.0	50.80						
306	69.4	27.30	-193.79	37,554.13	-235.889	55,643.57	42.1	1,772.41
2516	197.4	166.60						
2518	115.3	146.90						
Sum	2,368.7	1,014.4	157.6	166,375.3	(38.4)	177,289.0	196.0	42,097.1
Mean	263.19							
After Calibration with SAS								
Proj No.	Y(act)	Y(pred)	act-mean	(act-m)sq	pred-mean	(prd-m)sq	(act-predt)	(act-prd)sq
74	80.0	85.6						
75	912.0	657.3						
76	115.0	100.4						
77	523.0	343.8	259.81	67,501.81	80.61111	6,498.15	179.2	32,112.64
78	478.0	300.7	214.81	46,143.81	37.51111	1,407.08	177.3	31,435.29
79	432.0	311.6						
80	296.0	414.7						
81	164.0	155.2						
82	140.0	114.9	-123.19	15,175.50	-148.289	21,989.59	25.1	630.01
83	57.0	52.9						
306	69.4	70.5	-193.79	37,554.13	-192.689	37,129.01	-1.1	1.21
2516	197.4	303.5						
2518	115.3	125.4						
Sum	2,368.7		157.6	166,375.3	(222.9)	67,023.8	380.5	64,179.2
Mean	263.19							

Univariate Procedure

Variable=Residual

Moments

N	9	Sum Wgts	9
Mean	18.01111	Sum	162.1
Std Dev	113.0466	Variance	12779.54
Skewness	1.093243	Kurtosis	1.719849
USS	105155.9	CSS	102236.3
CV	627.6494	Std Mean	37.68221
T:Mean=0	0.477974	Pr>|T|	0.6455
Num ^= 0	9	Num > 0	5
M(Sign)	0.5	Pr>=|M|	1.0000
Sgn Rank	3.5	Pr>=|S|	0.7344
W:Normal	0.875725	Pr<W	0.1392

Quantiles(Def=5)

100% Max	254.7	99%	254.7
75% Q3	14.6	95%	254.7
50% Med	4.1	90%	254.7
25% Q1	-10.1	10%	-118.7
0% Min	-118.7	5%	-118.7
		1%	-118.7
Range	373.4		
Q3-Q1	24.7		
Mode	-118.7		

Extremes

Lowest	Obs	Highest	Obs
-118.7(	5)	4.1(	7)
-106.1(	8)	8.8(	6)
-10.1(	9)	14.6(	3)
-5.6(	1)	120.4(	4)
4.1(	7)	254.7(	2)

```
Stem Leaf                     #              Boxplot
   2 5                        1                 *
   2
   1
   1 2                        1                 *
   0
   0 011                      3              +--+--+
  -0 11                       2              +-----+
  -0
  -1 21                       2                 *
     ----+----+----+----+
 Multiply Stem.Leaf by 10**+2
```

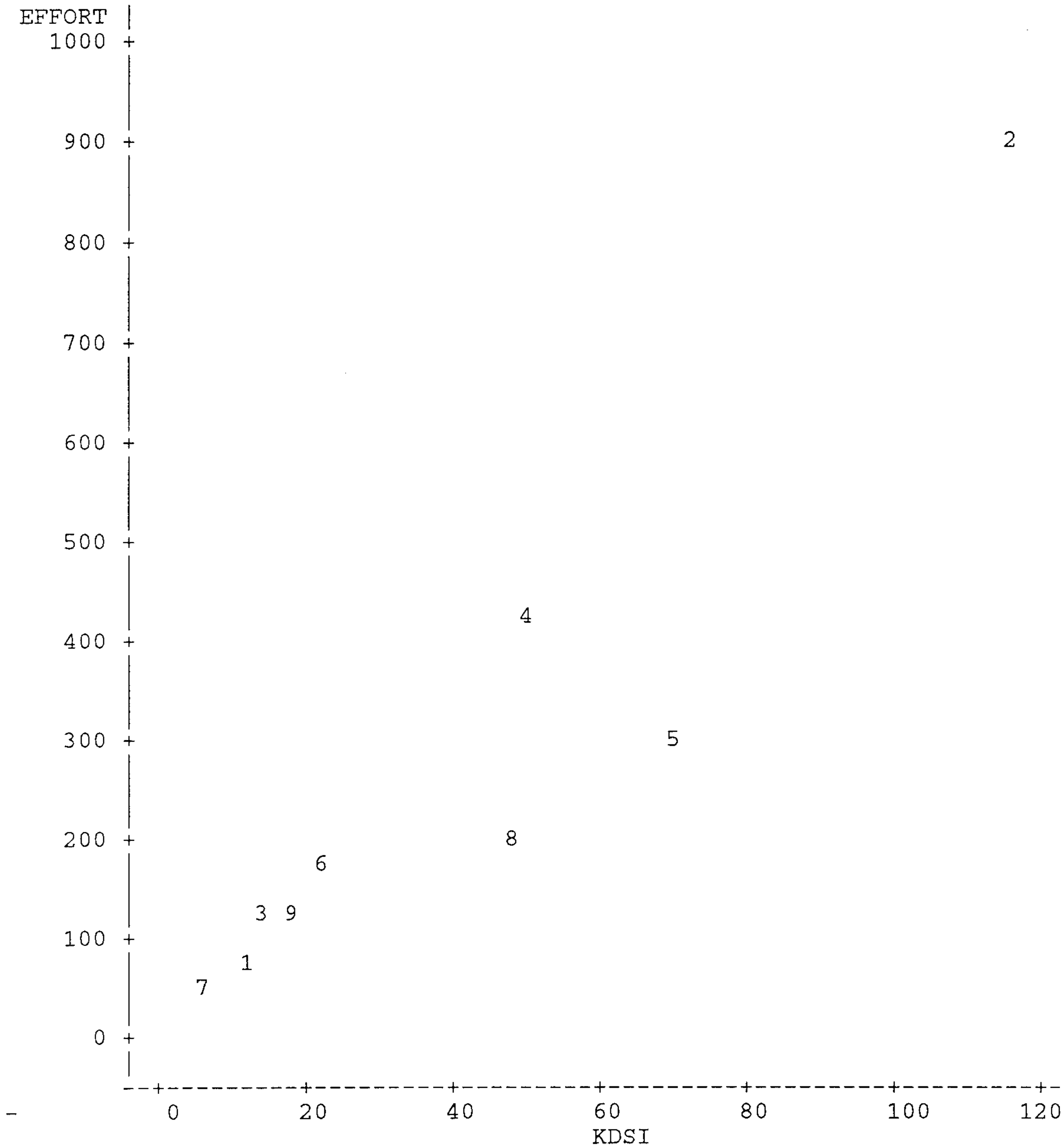
```
*--*    7/23/95   -    8:58:58 PM   *--*
                              AIR FORCE
                        INSTITUTE OF TECHNOLOGY
                        ***********************

CSC>ed space.dat
80 11.700 1.366
912 116.800 2.131
115 14.000 1.392
432 50.300 1.188
296 69.450 1.188
164 22.900 1.188
57 6.800 1.009
197.4 48.814 .684
115.3 18.004 1.001
[EOB]
*exit
CSC>ed spcnc.sas
filename unmanned'[bweber]space.dat';
OPTIONS LINESIZE=72;
data unmanned;
   infile space;
   input effort kdsi;
   leffort=log(effort);
   lkdsi=log(kdsi);
proc print;
   var effort kdsi leffort lkdsi;
   title 'Unmanned Space';
proc plot;
   plot effort*kdsi;
   plot leffort*lkdsi;
proc reg;
   model effort=kdsi/p clm cli;
     plot r.*kdsi;
   model leffort=lkdsi/p clm cli;
     plot r.*lkdsi;
[EOB]
*exit
```

Unmanned Space

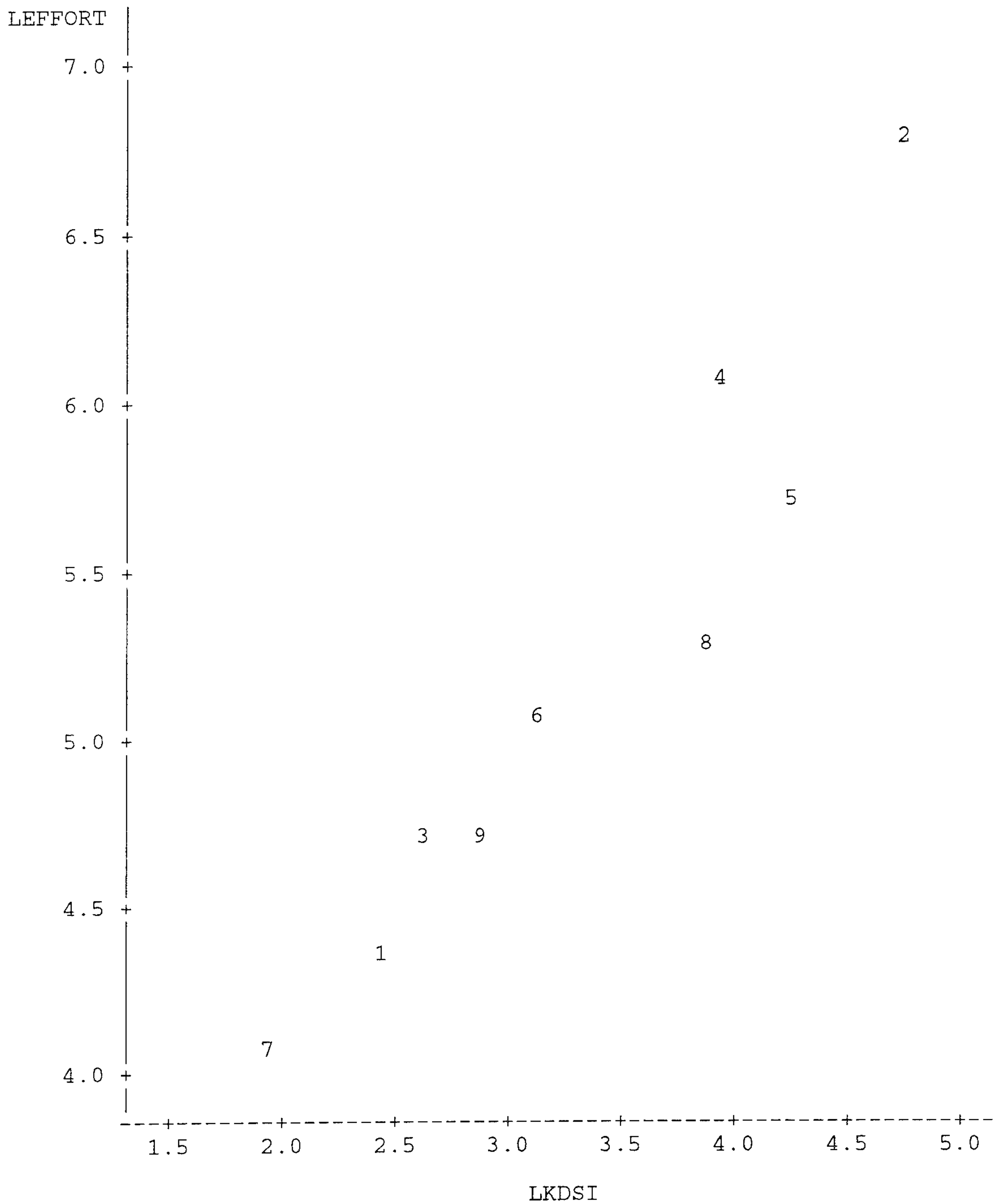
OBS	EFFORT	KDSI	LEFFORT	LKDSI
1	80.00	11.700	4.38203	2.45959
2	912.00	116.800	6.81564	4.76046
3	115.00	14.000	4.74493	2.63906
4	432.00	50.300	6.06843	3.91801
5	296.00	69.450	5.69036	4.24061
6	164.00	22.900	5.09987	3.13114
7	57.00	6.800	4.04305	1.91692
8	197.40	48.814	5.28523	3.88802
9	115.30	18.004	4.74754	2.89059

Plot of EFFORT*KDSI. Legend: 1 = 1st obs, 2 = 2nd obs, etc.



Unmanned Space

Plot of LEFFORT*LKDSI. Legend: 1 = 1st obs, 2 = 2nd obs, etc.



Unnamed Space

Model: MODEL1
Dependent Variable: EFFORT

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	514118.60703	514118.60703	51.061	0.0002
Error	7	70480.72186	10068.67455		
C Total	8	584599.32889			

Root MSE	100.34279	R-square	0.8794
Dep Mean	263.18889	Adj R-sq	0.8622
C.V.	38.12577		

Parameter Estimates

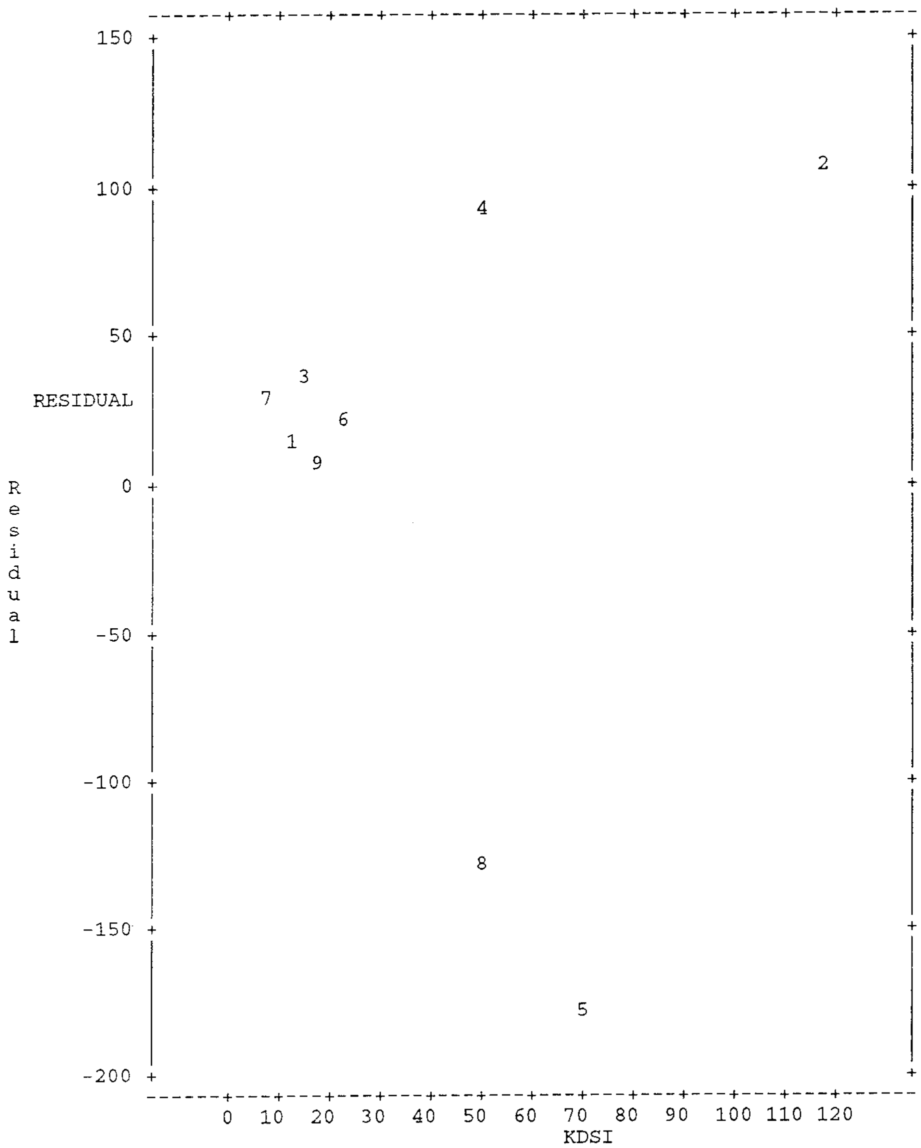
Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	-18.382864	51.68597661	-0.356	0.7326
KDSI	1	7.063467	0.98849024	7.146	0.0002

Obs	Dep Var EFFORT	Predict Value	Std Err Predict	Lower95% Mean	Upper95% Mean	Lower95% Predict
1	80.0000	64.2597	43.517	-38.6422	167.2	-194.4
2	912.0	806.6	83.082	610.2	1003.1	498.6
3	115.0	80.5057	42.099	-19.0428	180.1	-176.8
4	432.0	336.9	35.003	254.1	419.7	85.6149
5	296.0	472.2	44.431	367.1	577.2	212.7
6	164.0	143.4	37.415	54.8974	231.8	-109.9
7	57.0000	29.6487	46.764	-80.9311	140.2	-232.1
8	197.4	326.4	34.598	244.6	408.2	75.4320
9	115.3	108.8	39.820	14.6286	202.9	-146.5

Obs	Upper95% Predict	Residual
1	322.9	15.7403
2	1114.7	105.4
3	337.8	34.4943
4	588.2	95.0905
5	731.7	-176.2
6	396.6	20.6295
7	291.4	27.3513
8	577.4	-129.0
9	364.1	6.5122

Sum of Residuals	0
Sum of Squared Residuals	70480.7219
Predicted Resid SS (Press)	197450.8363

Unmanned Space



Unmanned Space

Model: MODEL2
Dependent Variable: LEFFORT

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	5.48967	5.48967	69.705	0.0001
Error	7	0.55129	0.07876		
C Total	8	6.04096			

Root MSE	0.28063	R-square	0.9087
Dep Mean	5.20856	Adj R-sq	0.8957
C.V.	5.38795		

Parameter Estimates

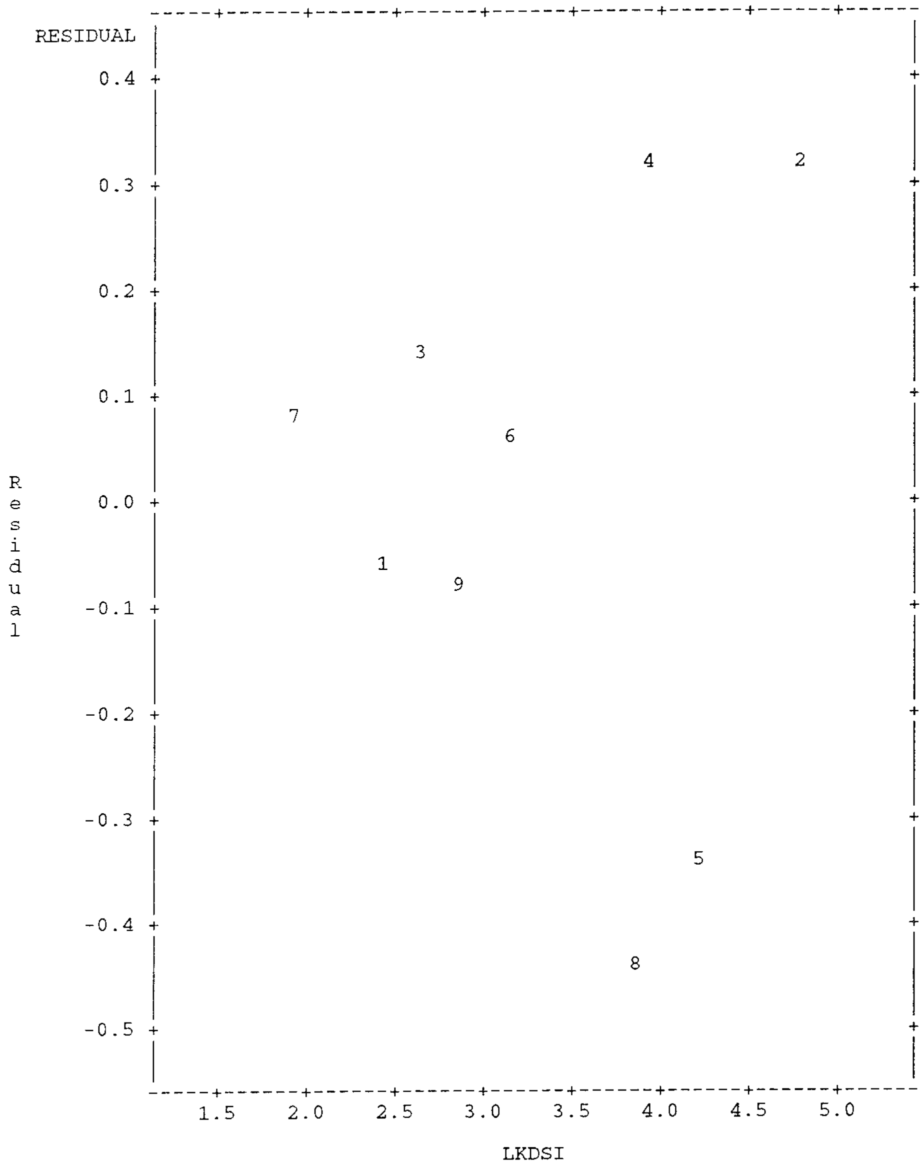
Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	2.270968	0.36407489	6.238	0.0004
LKDSI	1	0.885874	0.10610599	8.349	0.0001

Obs	Dep Var LEFFORT	Predict Value	Std Err Predict	Lower95% Mean	Upper95% Mean	Lower95% Predict
1	4.3820	4.4499	0.130	4.1415	4.7582	3.7181
2	6.8156	6.4881	0.180	6.0636	6.9127	5.7003
3	4.7449	4.6088	0.118	4.3299	4.8877	3.8890
4	6.0684	5.7418	0.113	5.4740	6.0097	5.0262
5	5.6904	6.0276	0.136	5.7071	6.3481	5.2907
6	5.0999	5.0448	0.096	4.8187	5.2708	4.3437
7	4.0431	3.9691	0.175	3.5542	4.3840	3.1865
8	5.2852	5.7153	0.112	5.4516	5.9789	5.0012
9	4.7475	4.8317	0.104	4.5861	5.0773	4.1241

Obs	Upper95% Predict	Residual
1	5.1816	-0.0678
2	7.2759	0.3275
3	5.3287	0.1361
4	6.4574	0.3266
5	6.7646	-0.3373
6	5.7458	0.0551
7	4.7518	0.0739
8	6.4293	-0.4300
9	5.5393	-0.0841

Sum of Residuals	0
Sum of Squared Residuals	0.5513
Predicted Resid SS (Press)	0.9769

Unmanned Space



Bibliography

1. Asbrand, D. 1993. "Re-engineering Demands Expert Management." Infoworld, 15: 51.

2. Babcock, C. 1994. "Software Development: TeamMate." Computerworld, 28: 6.

3. Boehm, B. 1981. Software Engineering Economics. Englewood Cliffs NJ; Prentice Hall.

4. Boehm, B. 1984. "Software Engineering Economics." IEEE Transactions on Software Engineering, SE-10: 4-21.

5. Bozman, J. 1993. "DoD Plots Common IS Scheme." Computerworld, 27: 10.

6. Brooks, F. 1987. "No Silver Bullet: Essence and Accidents of Software Engineering." Computer, 72: 10.

7. Coggins, G. And Russell, R.C. 1991. Software Cost Estimating: A Comparative Study of What the Models Estimate. MS thesis, AFIT/GCA/LAS/93S-04. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH (AD-A275989).

8. Conte, S. D., H.E. Dunsmore, and V.Y. Shen. 1986. Software Engineering Metrics and Models. Menlo Park CA; Benjamin/Cummings.

9. Daly, B.A. 1990. A Comparison of Software Schedule Estimators. MS thesis, AFIT/GCA/LSQ/90S-01. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH (AD-A229532).

10. Ferens, Daniel V. Class notes from COST 677, Quantitative Management of Software. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1994.

11. Ferrentino, A. 1981. "Making Software Development Estimates 'Good'." Datamation, 9: 236-238.

12. Kile, R. 1991. REVIC Software Cost Estimating Model User's Manual. Unpublished.

13. Mendenhall, W., D.D. Wackerly, and R.V. Scheaffer. 1990. Mathematical Statistics With Applications. Belmont CA; Duxbury Press.

14. Nordwall, B. 1994. "High-performance Software Lags Behind Hardware." Aviation Week & Space Technology, 140: 54-55.

15. Ourada, G. 1991. 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 (AD-A246677).

16. Schlotzhauer, S.D. and R.C. Littell. 1987. SAS® System for Elementary Statistical Analysis. Cary NC; SAS Institute Inc.

17. Shebalin, P. 1994. "Software Development Standards and the DoD Program Manager." Acquisition Review Quarterly, 1: 202-216.

18. Siegel, Joan, ed. 1992. Statistix®, Version 4.0, User's Manual. Tallahassee FL; Analytical Software.

19. Stukes, S. 1994a. Management Consulting & Research, Inc., Oxnard CA. Personal interview. 12 October 1994.

20. Stukes, Sherry and Henry Apgar. 1994b. Space and Missile Systems Center Software Development Database (Phase 5). Contract FO4701-90-D-0002 with Space and Missile Systems Center. Management Consulting & Research, Inc., Oxnard CA.

21. Thibodeau, R. 1981. An Evaluation of Software Cost Estimating Models. Huntsville AL. General Research Corporation.

22. Walker, W. 1978. An Approach to Software Life Cycle Cost Modeling. MS thesis, AFIT/GCS/EE/78-21. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH (AD-A064223).

Vita

Ms. Betty G. Weber is from Nashville, Tennessee. She graduated from the University of Tennessee in 1970 with a Bachelor of Science degree in Education and from Jacksonville State University in 1984 with a Master of Science degree in Education Administration. She is also a graduate of the U. S. Army's Basic Signal Officer's Course at FT Gordon, GA and the U.S. Army's Command and General Staff Officer's Course at FT Levenworth, KS.

Ms. Weber has been a civilian employee of the Department of the Army since 1981, having served as an education specialist on the staff at the U.S. Army Military Police School at FT McClellan, AL, the U.S. Army Signal School at FT Gordon, GA, and the U.S. Army Ordnance, Missile, and Munitions School at Redstone Arsenal, AL. Before arriving at AFIT, she had been employed, since 1990, as an operations research analyst by the Command Analysis Directorate at Missile Command (MICOM) at Redstone Arsenal, AL, providing matrix support to the program offices there and to the Cost Analysis Directorate at the Space and Strategic Defense Command (SSDC) in Huntsville, AL.

Upon graduation from the GCA program at AFIT, her follow-on assignment is expected to be as a member of the faculty at the U.S. Army Logistics and Management College (ALMC) at FT Lee, VA.

Permanent Address: 7205 Perrin Drive
Prince George, VA 23875

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1995	Master's Thesis

4. TITLE AND SUBTITLE
A CALIBRATION OF THE REVIC SOFTWARE COST ESTIMATING MODEL

5. FUNDING NUMBERS

6. AUTHOR(S)
Betty G. Weber, Civ., US Army

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology,
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCA/LAS/95S-13

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
SMC/FMC
2430 E. El Segundo Blvd. #2010
Segundo, 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 study examined the change in performance resulting from calibration of the REVIC software cost model and whether calibration improved the model's ability to estimate software development costs. Three calibrations were conducted on two operating environments using the SMC database. The calibrations consisted of one coefficient only and two coefficient and exponent calibrations, using different procedures. Analysis of the "goodness of fit" was accomplished using MRE, MMRE, RMS, RRMS, and the prediction test to measure the model's improvement in predicting ability. The Wilcoxon and Wilk-Shapiro tests were used to test for normality of the sample data. The author concluded that the REVIC model's predicting ability is insufficient when more than one independent variable is impacting cost. In the case of the two operating environments selected, military ground and unmanned space, the improvement in the estimating ability of the model, following calibration, was insufficient to predict cost in either the military ground or the unmanned space environment within an acceptable confidence level.

14. SUBJECT TERMS
Cost Estimates, Cost Models, Computers,
Software Cost Models

15. NUMBER OF PAGES
87

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	Unlimited