3-2001

Predicting Software Development Cost for Command and Control Systems

Anthony W. Fife

PREDICTING SOFTWARE DEVELOPMENT COST

FOR COMMAND AND CONTROL SYSTEMS

THESIS

Anthony W. Fife, Captain, USAF

AFIT/GAQ/ENV/01M-05

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

Wright-Patterson Air Force Base, Ohio

AFIT/GAQ/ENV/01M-05

PREDICTING SOFTWARE DEVELOPMENT COST FOR COMMAND AND CONTROL SYSTEMS

THESIS

Presented to the Faculty

Department of Systems and Engineering Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Acquisition Management

Anthony W. Fife, B.B.A, M.B.A
Captain, USAF

March 2001

AFIT/GAQ/ENV/01M-05

PREDICTING SOFTWARE DEVELOPMENT COST
FOR COMMAND AND CONTROL SYSTEMS

Anthony W. Fife, B.B.A., M.B.A.
Captain, USAF

Approved:

William K. Stockman (Chairman)	01 Mar 01 date
Edward D. White (Member)	1 Mar 01 date
Daniel V. Ferens (Member)	1 Mar 01 date

ACKNOWLEDGMENTS

I am grateful for the opportunity to attend the Air Force Institute of Technology (AFIT). It provided me with countless opportunities to meet many extraordinary people, gain wonderful friends, and experience rich times (both inside and outside of school). These are the things I will remember as I reflect on my time at AFIT.

I would be remiss if I didn't mention my advisor, Lt Col William Stockman, and my readers, Capt Edward White and Mr. Dan Ferens. I thank them for their time, promptings, and support they provided throughout the process of this thesis.

Probably the greatest thing that I relearned while I was here at AFIT is that advisors, instructors, classmates, bosses, and acquaintances come and go; but families are forever. I am truly thankful for my family. I am most thankful for my wonderful wife, [redacted]. She encouraged and supported me not only throughout this short thesis effort, but also through the adventures of my Air Force and college endeavors. She has been the glue to bind our family of three children together.

Anthony W. Fife

Table of Contents

Page

Page

List of Figures

Page

List of Tables

Page

AFIT/GAQ/ENV/01M-05

Abstract

Our nation is spending increasingly more dollars to develop software systems. Likewise, the Department of Defense (DoD) allocates more and more dollars to develop software. Consequently, the DoD needs to be able to accurately estimate what these systems' cost so they can develop reliable budgets. To estimate these software costs, the DoD relies on commercial software estimating tools. However, these tools are somewhat unreliable when it comes to estimating military systems, particularly Command, Control, and Communication Systems.

The purpose of this study was to develop a parametric model using linear regression to estimate software development costs for Department of Defense Command, Control, and Communications systems. The developed model is unique in a few ways. First, the model is derived from Department of Defense command and control data. Most other traditional models use a broader spectrum of data to create models, and then rely on calibration to tailor the model for a specific use.

Second, while traditional models require volumes of variables to create estimates, the developed model only requires a few key variables to estimate the amount of effort necessary to complete a project. The key variables were selected through analyzing common variables used in software cost estimating and performing regression analysis to focus in on the variables that have the greatest influence on expected effort.

PREDICTING SOFTWARE DEVELOPMENT COST FOR COMMAND AND CONTROL SYSTEMS

I. Introduction

Background

Estimating the cost of software systems is of great concern to not only the Department of Defense but also all those that come in contact with producing and procuring software systems. The reason for this is the shear magnitude of dollars being expended on software projects; which equal about 2.3% of our nation's Gross Domestic Product (Mannering, 2001). In the United States, industry and government spent 230 billion in 2000 on software projects (Mannering, 2001), compared with $70 billion in 1985(Hu et al, 1998: 144). Therefore, in this growing business sector that is costing more and more dollars it is important to be able to accurately estimate software costs.

Based on a survey done by Professors Lederer and Prasad, the great majority of respondents reported that their software estimates were dismal (only about one in four projects is completed at a cost reasonably close to the estimate)(DeMarco, 1995: 5). A 1984 study of seventy-two projects in twenty-three major U.S. companies revealed that the median cost overrun is about 34 percent with and average of 67 percent, and the average schedule slip is about 22 percent (Hu et al., 1998: 144). Furthermore, it's not uncommon to see large software projects with 200 to 300 percent cost overruns and 100 percent schedule slips (Hu et al., 1998: 144). Hu, Plant, and Hertz believe one of the top contributors to these massive overruns is inaccurate estimation of development cost and schedule, which lead to unrealistic expectation and project planning (Hu et al., 1998).

General Issue

Currently the DoD relies heavily on commercial software estimating tools. Some of the more popular models used by the DoD include the following: PRICE-S, SEER-SEM, and SLIM, CHECKPOINT, COCOMO II, and SAGE. These parametric models are popular because they are so relatively easy to use and even a novice estimator can quickly begin estimating programs. However, the ease of use comes with a price, the models don't seem to be very accurate in estimating DoD software programs (Ferens, 1996: 29-31).

Because of the inaccuracies of these commercial models, some suggest the models may be made more useful through calibration (Ferens and Christensen, 1997). Ferens and Christensen state, "a solution to the accuracy problem may be to calibrate the models to the user's environment (Ferens and Christensen, 1997, 43)." Between the years 1994 to 1997 Air Force Institute of Technology (AFIT) masters students calibrated nine different popular models. The AFIT student research concluded that there are mixed results when popular models are calibrated with military data. Some models are more suitable than other models depending on their application, environment, and type of software being developed (Ferens and Christensen, 1998).

Specific Issue

A significant area for DoD software estimating is Battle Management Command, Control, and Communication (BMC3). BMC3 software is the component of weapon system software that communicates, assimilates, coordinates, analyzes, interprets information, and provides decision support for military commanders. It provides instantaneous situation assessment, allowing for advantageous timely position and decision making (Cummings et al., 1998). The OSD CAIG has seen a significant increase in the complexity and size of BMC3 suites on a variety of weapons systems. Software development constitutes the majority of the effort in these BMC3 suites (system code counts range from 1 Million SLOC to 3 Million SLOC).

In light of the increasing quantity and size of these systems, it is even more important to be able to accurately estimate them. However, according to the AFIT studies mentioned above, none of the calibration efforts effectively improved the commercial model's accuracy in estimating BMC3 systems (Ferens and Christensen, 1998). The reason for this is that BMC3 systems are significantly different from other systems. The Naval Center for Cost Analysis (NCCA) explains that the systems are more software dependent than non-BMC3 systems. Consequently, their productivity of software development efforts may be lower than traditional military systems. (Cummings et al., 1998) Therefore, the purpose of this study is to examine Battle Management Command, Control, and Communication (BMC3) software development efforts to determine key cost drivers and establish appropriate cost estimating relationships for these types of systems.

As part of this study, the software development environment for historical systems must be taken into account. Specifically, size of the project, a variety of complexity variables, the schedule, differing types of programming languages, and other appropriate historical BMC3 suite data will be examined.

Research Approach

This research, through relevant literature and software cost models, will reveal insights concerning approaches traditionally attempted to estimate similar software development efforts. Data will be collected from historical systems that are similar to BMC3. The Space and Missile Systems Center (SMC) and the Electronic Systems Center (SMC) software databases contain the similar types of systems that were used in this analysis. The gathered data was analyzed using multivariate regression. Through a review of the data and detailed analysis, the cost estimating relationships for BMC3 systems are described.

Research Contribution

This research is unique to recent efforts at the Air Force Institute of Technology. Over the past few years, AFIT's software estimating focus has been on calibrating commercial models. These commercial models were built using commercial databases, and then calibrated with military data to fit the military's needs. Conversely, this effort will analyze military data to build a military model to fit military needs. The results of this research effort will provide analysts needed insight into the life cycle costs associated with the development of BMC3 systems and provide them with a set of cost estimating relationships that will enable them to adequately cost future development efforts for BMC3 systems.

Scope of Research

The purpose of this research is to determine if development costs can be accurately predicted using a model that is derived solely from DoD command and control data. No data points other than those provided from the DoD sources mentioned above will be used in the research. Additionally, the model derived from this research is intended to estimate Command and Control projects only. Once again, the purpose of this model is to determine if a more accurate and useful model can be built using specific data that relates directly to the software program application, BMC3.

Thesis Overview

This chapter provides an overview to the problem DoD faces in estimating BMC3 software systems. It is questionable whether commercial models, either "as is" or calibrated, fulfill the accuracy requirements desired by the DoD (Ferens and Christensen, 1997). Bad estimates increase the frustration and tension among contractors, program managers, and estimators. Contractors overrun budgets, program managers set unrealistic goals, and estimators get labeled the "bad guys" because they create inaccurate estimates. Therefore it's clearly necessary that something needs to be done to create more accurate estimates.

Chapter II, the Literature Review, provides a summary of the current thinking in the industry concerning software cost estimating. The chapter discusses important ideas and components to software cost models. The ideas include the basic form that many of today's model use. The components section include a description of the key cost driver that the literature feels influences the level of effort necessary to develop software systems.

Chapter III, Methodology, details the steps of collecting and scrubbing the data, building the model through statistical techniques, and validating the proposed model.

Chapter IV, Findings, describes the results and findings of the model building effort in Chapter III. Included in this chapter is the complete model and the effects each of the significant attributes have on the model.

Chapter V, Conclusions and Recommendations, reviews the findings of Chapter IV and determines to what extent the goals set out previously in this chapter are met. Additionally, through the conclusions, it is apparent where further research should be done. Recommendations concerning future efforts are also included in this chapter.

II. Literature Review

Introduction

The purpose of this chapter is not to give an all-inclusive explanation of software estimating. There is a multitude of books, like T. Capers Jones' *Estimating Software Costs* or Barry W. Boehm's *Software Engineering Economics*, which address this very subject. Still, this chapter will cover the basics of software cost estimating. To understand software estimating models it's important to understand difficulties involved in estimating and why estimating is necessary. It's also important to understand some of the key variables and assumptions that other models include in their algorithms. Then with knowledge of the key variables and assumptions, the creation of software models will make more sense. This chapter discusses how and what statistical models are used to create software-estimating tools. Additionally, this chapter discusses past efforts to improve effectiveness of software estimating (specifically in the area of Command and Control software systems).

Software Cost Estimating Models

Stutzke in his article, *Software Estimating Technology: A Survey,* explains there are two basic classes of estimation methods: experience-based estimation and parametric models. Experience-based models rely on estimator's knowledge and experience in the field; however, the weakness of this methodology is that the estimator may not correctly recall or apply the things he knows. Conversely, parametric models are models that are based on historical data. Because they are based on historical data, they tend to have a particular "perspective." If the data is slanted towards commercial or military data or a particular application, the perspective of that model will have a commercial, military, or a specific application's flavor. Consequently, when using parametric models, estimators should find the model that best fits the type of project they are estimating or make sure the model is properly calibrated for that particular environment (Stutzke, 1996).

Boehm expands Stutzke's two basic categories to seven methods of software cost estimation. The seven methods are described in Table 1.1.

Table 2.1. Strengths and Weaknesses of Popular Models (Boehm, 1981: 329-338)

Method	Description	Strengths	Weaknesses
Algorithmic Models	These methods provide one or more algorithms, which produce a software cost estimate as a function of a number of variables, which are considered to be the major cost drivers.	Objective, repeatable, efficient, able to support sensitivity analysis.	Model's data may not be representative, may not account differences.
Expert Judgment	This method involves consulting one or more experts, perhaps with the aid of an expert-consensus mechanism such as the Delphi technique.	Quick, able to factor differences, such as new techniques or architectures.	May be biased; either optimistic or pessimistic.
Analogy	This method involves reasoning by analogy with one or more completed projects to relate their actual cost of an estimate of the cost of a similar new project.	Estimate is based on actual experience on a project.	Project may not be representative of the estimated project.
Parkinson	A Parkinson principle ("Work expands to fill the available volume") is invoked to equate the cost estimate to the available resources.	Lots of bells and whistles (if your into that sort of thing).	Not particularly accurate and supports poor practices.

Price-to-Win	The cost estimate developed by this method is equated to the price believed necessary to win the job (or the schedule believed necessary to be first in the market with a new product, etc.).	Wins contracts.	Purely subjective, based on what the customer wants to hear.
Top-Down	An overall cost estimate for the project is derived from global properties of the software product. The total cost is then split up among the various components.	Includes all system level requirements like integration, training, and manuals.	Low-level technical difficulties are over looked. Components may be left out.
Bottom-Up	Each component of the software job is separately estimated, and the results aggregated to produce an estimate for the overall job.	Looks at each individual component and errors tend to balance each other out.	May overlook system level requirements like integration, training, and manuals.

The key for estimators is to determine which of these methods is most useful in their particular situation. Each method has its particular strengths and weaknesses that may be capitalized on in a specific situation and often the models compliment each other (Boehm 1981: 341). Still, it's important to note that not all models may produce objective estimates. Boehm feels the Parkinson and the Price-to-Win methods don't produce good objective measures of the effort required for developing software (Boehm, 1981: 341).

In the beginning stages of a program, when little is know about a program's requirements, expert judgement and rules of thumb may be most useful in estimation. Then, when the requirements become more established, more advanced techniques may be employed.

Model Building

Throughout this paper, the focus of estimating models will lean more towards parametric or algorithmic models that are based on historical data. Parametric models are popular because they are relatively easy to use; even novice estimators can quickly begin estimating programs.

Literature describing the actual techniques for building software cost estimating models is somewhat sparse for such a mainstream activity. The reason for this is that most of the commercial software costs estimating tool vendors regard their estimating methods and algorithms as trade secrets (Jones, 1998: 20). Nevertheless, we have a general idea of what most models look like. According to Caper Jones in his book, *Estimating Software Costs*, most software estimating models follow a form similar to the one illustrated in Figure 2.1.

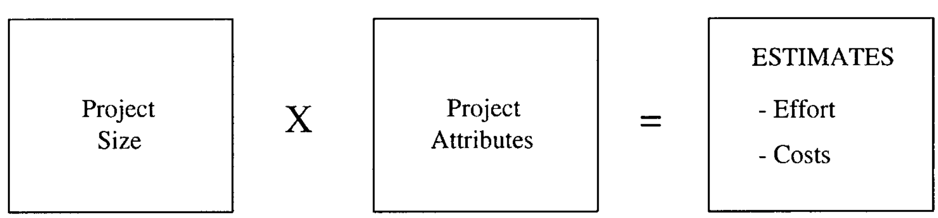


Figure 2.1. Basic cost estimating model (Jones, 1998: 6)

Program Size Overview

The first block in Figure 2.1, project size, is traditionally measured and reported in one of two ways. The first way to measure size is counting lines of code and the second way to measure size is by counting function points. Both counting methodologies are described in the following paragraphs.

Probably the most significant and important piece of data to collect is the size of a software development program. Size is important because it is usually the key variable in most estimation models. Nevertheless, one should be careful when talking about size because there are various ways to measure the size of a computer program. Two of the more popular methods are counting Source Lines of Code (SLOC) and measuring Function Points. Both methods are currently used within the software industry, with Function Points being the newer of the two measures. Size is relatively easy to compute and therefore is a popular if not necessary component for software cost models. Conte states that size is probably the most important factor for many software development models. He also explains that size is also important for developing a secondary factor, productivity (Conte, 1986: 32). Productivity is a factor of size divided by the effort to build a software project, which results in a number that describes the number of man-hours to complete a line of code.

Function Points. Of the two sizing methodologies, Function Point sizing is the newer of the two and less commonly used within the Department of Defense (DoD). Still, there are some avid proponents of Function Points that believe it is a superior method of sizing software systems. Function Points are based on external attributes of a software project, which consists of the following five primary elements: (1) external inputs, (2) external outputs, (3) external inquiries, (4) internal logical files, and (5) external interfaces (Jones, 1998: 303).

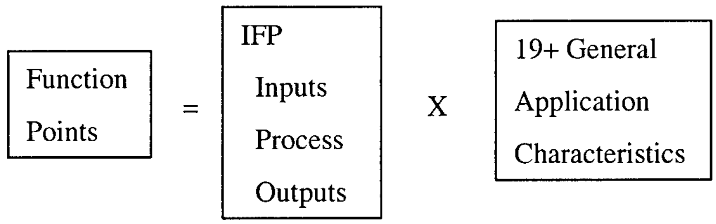


Figure 2.2. Components of Function Points (Symons, 1991: 22)

Function points can be extremely difficult to calculate, but are easy to understand because the measured attributes are externally apparent. Charles Symons explains that Function Points are a combination of information processing size (IPS) and general application characteristics (Symons, 1991). Figure 2.2 illustrates Charles Symons' basic structure for calculating Function Point size.

Function Points are an interesting concept and are gaining wider acceptance in the estimating community. Still, the Department of Defense (DoD) hasn't jumped on board the Function Point bandwagon. Most of the models used by the DoD and most of their databases are SLOC based. This may be something the military may want to look into in the future; Jones contends that because of the completeness of military specification, DoD projects would be ideal to estimate using Function Point sizing methods.

Source Lines of Code (SLOC). Even though it may seem simple to simply count the SLOC, it is more complex than it appears. The problem with counting SLOC is that not everybody agrees what qualifies as a line of code (Conte, 1986: 32). An uncomplicated solution would be to simply count the lines or carriage returns; however, blank lines and comment lines probably shouldn't be included in the count. Within industry there are two distinctly different SLOC counting methods: physical and logical.

Physical SLOC counting is simply counting the number of carriage returns. Logical SLOC is determined by counting logical units (for example, an IF-THEN-ELSE statement is considered a logical unit). The methodology employed may make a significant difference. An Institute for Defense Analysis (IDA) study concluded that physical code counts are generally about 20% higher than logical code counts (Cummings et al., 1998). Within the industry, most researchers agree blank lines and comment lines shouldn't be included. If they were included, analysts could easily inflate the size of the software program (Conte, 1986: 34). The following is a definition for lines of code that is commonly accepted throughout the industry:

> A line of code is any line of program text that is not a comment or blank line, regardless of the number of statements or fragments of statements on the line. This specifically includes all lines containing program headers, declarations, and executable and non-executable statements. (Conte, 1986: 35)

Still, with this definition in hand there is some ambiguity concerning how or what to count when counting SLOC. Even with all this definitions, counting logical SLOC can be difficult because much of the count may be left up to interpretation.

There are many compelling reasons why SLOC is a widely used metric. SLOC metrics are relatively easy to count (easier for physical lines of code). Line of Code measurements can easily be mathematically converted to another sizing methodology, including function points. Additionally SLOC is the most popular metric used in many of today's commercial software estimating tools (Jones, 1998: 319).

Reuse of Code and Effective Size. The Naval Center for Cost Analysis comments that not only knowing the amount of source code necessary, but also knowing the "condition" of the code is also important (Cummings et al., 1998). Many developed software systems aren't built from the ground up. There may exist projects or programs that are used to create new software systems. In determining the size of a software system it simply wouldn't be correct to consider reused code the same as newly developed code. Still, reused code simply doesn't come free of charge without any added effort. To account for the size of projects using both new and reused code a unique measure is employed called "effective sizing" or "equivalent sizing." The equivalent SLOC (ESLOC) takes into account the fact that reused code doesn't take the same amount of effort to put into a program as new code. One method for determining the effective size of programs uses an Adaptation Adjustment Factor (AAF) that is based on engineering judgment of distributed effort between percent design modification (DM), percent code modification (CM), and percent integration and test modification (IM). For example the AAF may appear as follows:

$$AAF = 0.4DM + 0.3CM + 0.3IM \qquad (2.1)\ (\text{Boehm, 1981})$$

In the example above the design requires a 40 percent redesign, code requires 30 percent redesign, and test requires 30 percent redesign. After calculating AAF, ESLOC is calculated using the following formula:

$$ESLOC = \text{New SLOC} + (AAF * \text{reused SLOC}) \qquad (2.2)\ (\text{Boehm, 1981})$$

The biggest drawback with the method explained above is that engineers aren't infallible when estimating the percentages for DM, CM and IM. Consequently, size estimates for reused pieces of code may only be as good as the best guesses of your best engineers.

Project Attributes

The second block illustrated in Figure 2.1, project attributes, may include (but not restricted to) the following information:

1. Rate at which a project's requirements may change
2. Developing team's experience with this kind of project
3. The standards that will be employed, i.e. ISO, DoD
4. Programming languages utilized
5. Programming processes or methods
6. Reusable code
7. Development tools used
8. Office dynamics/environment
9. Schedule pressure (internal or external)
10. Complexity of the project

(Jones, 1998: 6)

Then, by factoring project size and attributes, one can estimate a software project's schedule, effort, costs, and deliverables.

Next to size, the most important data for software models are the projects attributes. These are the characteristics that make the development project unique. Each software development effort has special needs or attributes that will either increase or decrease the amount of effort necessary to complete a project. For example, it makes sense that a project that is inherently more difficult than normal will take more effort to complete.

An important attribute that seems to make a difference is the programming language used to develop the project. This seems especially true when the projects are written in second-generation languages (2GLs) versus third-generation languages (3GLs). After all, 3GLs were developed to make writing and understanding programs easier (Cummings et al., 1998). The reason for the difference in ease of use is that 2GL languages are one step above machine language and are awkward to use, while 3GL languages and above are closer to spoken language and easier to write and understand Cummings et al., 1998). Assembly language is the primary 2GL language utilized by past DoD projects. The NCCA completed a study to determine if productivity levels are affected by whether a project is written in 2GL or 3GL. They found there is a significant productivity difference between the two generations of language. However, they found that no significant difference occurred between different 3GLs(Cummings et al., 1998). Consequently, it seems estimators need to pay particular attention when considering differences in language generation.

There are a number of other variables that are believed to have a significant effect on the amount of effort used to develop software. These variables fall into four basic categories; which include personnel, technology, processes, and environment (Jones, 1998: 7-8). The levels of experience the personnel have seem like a common sense factor. The personnel category may include factors like programming experience, language experience, or operating environment experience. As with everything else in our lives, technology may significantly influence how things are done. For example, technology drives whether automated tools or manual methods are used in writing code. Of course, programmers using automated tools would be expected to have a higher productivity rate than those not using them (Jones, 1998: 7). It's also obvious that the processes a team uses will affect their productivity. On the other hand the environmental influence on productivity is not as obvious. Environmental factors include where people work, and the relations they have with those around them. Jones states, "surprisingly, access to a quiet, noise-free office environment is one of the major factors that influences programming productivity" (Jones, 1998: 7).

Another item that should be considered is the type of application the software provides. Military applications are notably different than commercial applications. Additionally, there is a lot of variety within military application; whether the software is used in an aircraft, ship, or in space may make a significant difference (Jones 1998, 99-100). Furthermore, of all the military applications one seems to stand out, specifically Command and Control systems. The Naval Center for Cost Analysis (NCCA) explains that Command and Control systems are more software dependent than non-BMC3 systems. Consequently, their productivity may be lower than traditional military systems (Cummings et al., 1998).

Algorithmic Models

Models that employ various algorithms derive their algorithms from statistical techniques. Regression analysis is a method used to determine the relationship of dependent variables and independent variables. In the case of software development, the dependent variable is the level of effort to develop computer programs and the independent variables are the drivers that influence the level of effort necessary for development. According to Conte, Dunsmore, and Shen a large number of models, both linear and nonlinear, have been proposed for effort estimation (Conte et al., 1986: 279). The following paragraphs will review both types of statistical models

Linear Statistical Models. Linear models are popular because they employ equations are simple to understand and use relationships that are relatively easy to explain. The basic form of a linear statistical model is illustrated in equation 2.3.

$$E = \beta_0 + \sum_{i=1}^{n} \beta_i x_i \qquad (2.3)$$

The dependent variable, *E*, is the amount of effort necessary to develop software. The x_i are the factors or attributes believed to affect effort. The attributes or factors include things like size, schedule, personnel experience, complexity of the project, and many of the other items mentioned in previous sections (Conte et al., 1986: 279-280).

Systems Development Corporation (SDC) developed a linear model by looking at 104 different attributes and then narrowed them down to 14 key attributes. SDC's 14 attributes include the following: lack of requirements, stability of design, percent math instructions, percent I/O instructions, number of subprograms, programming language, application, stand-alone program, first program on a computer, concurrent hardware development, random access device used, different host, number of personnel trips, and military development (Conte et al., 1986: 280-281).

As shown above with the SDC model, literally hundred of attributes may affect effort. Many of these variables may account for the same thing and therefore may be accounted for in a single variable. Thus, the volumes of variables may be reduced to a smaller subset of variables that tell the same story. Still, one should use caution when using or interpreting a linear model. For example, the individual terms, x_i, and their coefficients should not be interpreted independent of other terms in the model's equation. All the terms act in concert with each other to predict the estimated effort (Conte et al., 1986: 280). Additionally, one should use caution when attempting to estimate a project whose attributes or expected effort is outside the range of the model's attributes (Conte et al., 1986: 280).

Nonlinear Statistical Models. Conte, Dunsmore, and Shen reveal that most nonlinear models they've studied take on the basic form illustrated in equation 2.4.

$$E = (a + bS^{c})m(X) \qquad (2.4)$$

As with the linear model the dependent variable, *E*, is the amount of effort necessary to develop software. The S is the estimated size of the project, usually expressed in SLOC; *a*, *b*, and *c* are constants derived by regression analysis; and *m(X)* is an adjustment multiplier that depends on one or more attributes denoted by the vector *X* (Conte et al., 1986: 281).

The problem with nonlinear models is that *m(X)* may be a very complicated function of several variables. Consequently nonlinear models tend to be harder to understand and more difficult to explain, especially the relationship of the cost drivers effect on effort. Still, most commercial models used by the DoD at least partially employ some type of nonlinear statistical model (Conte et al., 1986: 300). Additionally, nonlinear regression is too complex to lend itself to standard regression analysis techniques. Instead, it is more customary to have a general idea of the form of the model, or a baseline, and then adjust the model to fit an application's particular needs (Conte et al., 1986: 282).

Commercial Models and Calibration Efforts

Commercial Model Background. Because of the explosive growth in the software industry there has been an equally explosive growth of software estimating packages. Both organizations that procure software and those that produce software have a need to know the cost of developing and producing finished software systems. As of 1998 there are at least 50 commercial software-estimating tools (Jones 1998: 37). It would be interesting to analyze how each commercial software model estimates development costs. Specifically, it would be interesting to analyze the algorithms the models employ and how they account for various peculiarities of each development project. With this kind of insight, model builders could build new models on the shoulders of those that have been built before. Unfortunately there isn't a whole lot of insight into how many of the commercial models were built because the algorithms are considered trade secrets and the databases the models were derived from are usually proprietary (Jones, 1998: 20). According to Capers Jones, most of these commercial estimating tools share the following same basic features:

1. Database containing hundreds of thousands of software projects
2. Can perform size predictions
3. Automatically adjust estimate based on tools, languages and types of products
4. Predict quality and reliability
5. Can predict maintenance and support costs
6. Predict and help prevent problems

(Jones, 1998: 5)

Still, Jones states that because of military unique practices and characteristics in developing and procuring software many of the commercial tools that were developed using non-military data-points are incapable of accurately estimating military projects without calibrating the tools (Jones, 1998: 38). Additionally, Ferens and Christensen commented in a recent article, "while these models (commercial models) are sophisticated, they do not always produce accurate results, especially in the DoD environment where organizations contract with numerous and diverse software development companies" (Ferens and Christensen, 1997, 43).

Calibration Background. There have been many efforts to calibrate existing commercial software models to make them useful in estimating military software projects. Ferens and Christensen state, "a solution to the accuracy problem may be to calibrate the models to the user's environment" (Ferens and Christensen, 1997, 43). As a direct result of Ferens' and Christensen's belief that calibration may help commercial models estimate more accurately, starting in 1994 and finishing in 1997, masters students at the Air Force Institute of Technology (AFIT) initiated a study of calibrating traditionally used models. Over the next few years AFIT calibrated nine different commercial models (PRICE-S, REVIC, SASET, SEER-SEM, and SLIM, SOFTCOST, CHECKPOINT, COCOMO II, and SAGE). Data from the Air Force's Electronic System Center (ESC) and Space and Missiles System Center (SMC) were gathered and used to calibrate the above-mentioned models. The data were stratified into various software development categories (including unmanned space programs, military avionics programs, military ground command and control programs, military mobile programs, missile programs, military ground signal processing programs).

The SMC software database contains fields for over 50 items for each program in the database, which include:

- General information: software level, operating environment, software function, development standard, contracting agency, and type of contract
- Cost, size, and schedule information: effort in person-months, phases included, size (in SLOC), schedule and database size

- Software characteristics: 28 items, including complexity level, language, code mix, development method, and environmental factors. Most of these characteristics may be used as inputs to the REVIC, SEER-SEM, and PRICE-S models.
- Maintenance information: Number of years, effort, quality, documentation, and number of lines of code added, deleted, and modified.

(Stukes et al., 1999)

The following is a basic outline of the methodology the AFIT master students used by the last 5 theses efforts in calibrating the various models:

- if the data set has 8 or fewer, use all for calibration
- 9 to 11 data points, use 8 for calibration and remainder for validation
- if 12 or more data points, use 2/3 for calibration and 1/3 for validation

Statistical criteria were used to determine whether the model accurately estimates actual costs. This criteria was used both to determine if the stand-alone model is useful and if calibrated model is useful. Then one could see if the calibration improves the effectiveness of a model. The following criteria, which was proposed by Conte, Dunsmore, and Shen in their book, *Software Engineering Metrics and Models*, (Conte et al., 1986: 172-176), was utilized by the AFIT students:

$$\textit{Magnitude of Relative Error (MRE)} = |\textit{estimate - actual}|/\textit{actual} \quad (2.5)$$

$$\textit{Mean Magnitude of Relative Error (MMRE)} = (\textit{MRE}) / n \quad (2.6)$$

$$\textit{Root Mean Square (RMS)} = [(1/n)\,(\textit{estimate - actual})^2]^{1/2} \quad (2.7)$$

$$\textit{Relative Root Mean Square (RRMS)} = \textit{RMS} / [(\textit{actual})/n)] \quad (2.8)$$

$$\textit{Prediction Level (Pred (.25))} = k/n \quad (2.9)$$

In the above equations, n is the number of data points in the sub set and k is number of data points with *MRE* less than .25. According to Conte, et al., a model exhibits a "good" fit when the following criteria are met when the models' estimated effort is compared with actual effort for a data set:

- The *MMRE* is less than 0.25.
- The *RRMS* is less than 0.25.
- The predicted level (or estimated effort) is within 25% of actual effort at least 75% of the time.

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

Calibration Results. From the AFIT studies mentioned previously, it was concluded that there are mixed results when commercial models are calibrated with military data. Some models are more suitable than other models depending on their application & environment & type of software being developed. Because this thesis effort is directed towards the Command and Control area of these calibration efforts, the following table shows the results of each model for Command and Control applications. Some of the nine models are not included in this table because there wasn't sufficient data to calibrate the models for the command and control application.

Table 2.2. Results Of Calibrating And Testing Popular Models

Author (Year)	Cost Model	MMRE	RRMS	Pred (0.25)	MMRE	RRMS	Pred (0.25)
Kressin (95)	SLIM	0.62	n/r	0.00	0.67	n/r	0.00
Rathmann (95)	SEER-SEM	0.53	1.03	0.31	0.31	0.30	0.29
Mertes (96)	CHECKPOINT	0.19	0.15	0.50	0.17	0.16	0.50
Marzo (97)	SAGE (SMC)	0.40	0.59	0.37	0.35	0.56	0.41
Marzo (97)	SAGE (ESC)	0.38	0.68	0.27	0.37	0.53	0.22

As shown in Table 2.2, none of the programs listed above passed the criteria as established by Conte, Dunsmore, and Shen in their book, *Software Engineering Metrics and Models*. Consequently, none of the commercial products, whether they are calibrated or not calibrated, adequately estimate Command and Control systems based on the criteria.

Conclusion

This chapter has illustrated many of the problems and difficulties as well as the necessity of accurate software cost estimating. The discussion included what the commercial sector is doing in the field of software cost estimating, especially with commercial models. It was shown how these commercial models might be calibrated, but the calibration didn't significantly improve the accuracy of military estimates. Because the commercial models don't satisfactorily estimate software costs, especially in the area of Command and Control, it's necessary to look to other sources for software estimating. This chapter discussed how software models are created from historical data using statistical techniques. The historical data includes the key cost drivers that influence the level of effort necessary to develop software systems. With these key variables, a new model may be developed that is based totally on like systems and hopefully the accuracy will be of an acceptable level. After all, it makes sense that models that use general information will generally get you in the ballpark; possibly models that use specific information will get ball over the plate.

III. Methodology

Methodology Introduction

The purpose of this methodology is to build a model that may accurately predict the level of effort necessary to develop BMC3 software systems. Past efforts in this area have involved calibrating existing popular models with applicable data that is relevant to the type of project being estimated. Regression analysis is a technique that is used to determine mathematical relationships that may potentially be used to predict future responses. In the simplest form, data is analyzed through regression analysis, which creates a model that may be used to determine a predicted answer as illustrated in figure 3.1.

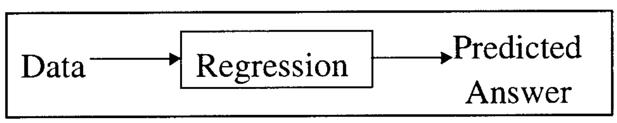


Figure 3.1. Linear Regression Illustrated

This methodology will include selecting and preparing historical data, performing regression analysis on that data, and using the regression to predict the development cost/effort of software development projects. After the model is created, the results will be tested and validated to determine its usefulness. Figure 3.2, the process flow diagram, illustrates this process and acts as an outline for the remainder of this chapter.

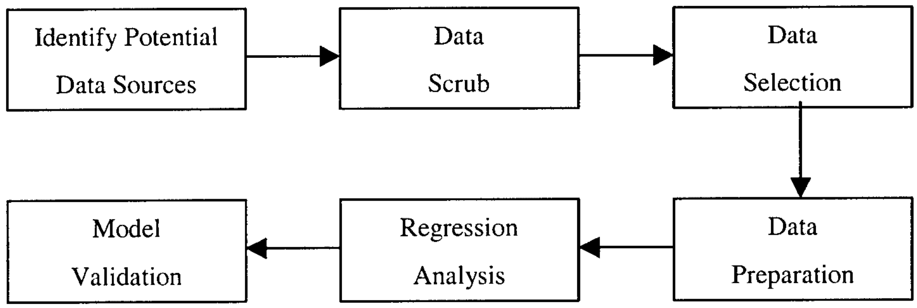


Figure 3.2. Methodology Process Flow Diagram

Identify Potential Data Sources

The Data for this study comes primarily from two sources, Electronic Systems Center (ESC) and Space and Missiles System Center (SMC). In the past, both centers had on going projects to collect software attributes data. The attributes the centers collected are based on attributes used by used software estimating models like SEER-SEM, PRICE-S, COCOMO, and other popular models.

SMC Data Base. The Space and Missile Center Software Database (SWDB) was developed to access and display data collected and stored by SMC. The SWDB was developed under the direction of the USAF Space and Missile Systems Center, with assistance from the Space Systems Cost Analysis Group (SSCAG). The SWDB currently contains almost 2500 data records. Each of the records contains up to 276 of the development effort's attributes (Stukes & Nguyen, 1999).

In the past, the SWDB was maintained by a contractor, however, SMC decided not to fund the contractor that compiled the database (Carpio, 2000). The majority of the records in the SWDB are inadequate in terms of completeness of information. Of the 276 fields for any given record, only a portion of the fields may be complete. Additionally, the projects and the contractors involved in each project are masked, which makes it impossible to verify the validity of the data. Consequently, only a small portion of the 2500 records is useful.

ESC Data Base. The ESC database is comprised of 52 separate projects. These projects include complete software projects or software sections of larger projects. When the sections are broken out of the 52 projects, there are 169 separate records of software attribute data. The collection of ESC's data dates back to 1974 when the United States Air Force (USAF) and MITRE joined forces to build the database. Recently ESC's software database keeping and maintenance was turned over to Tecolote Research, Inc. Tecolote migrated the data base from an Microsoft© Excel format to the Automated Cost Data Base (ACDB) format that is included as part of the cost estimating relationship library embedded in ACEIT©, a cost estimating and modeling tool created by Tecolote. This migration makes it easier for estimators to use the data while estimates are created in ACEIT©.

Unlike the SMC database, the ESC database is currently being used and maintained. Consequently ESC's data appears to be more useful because of the completeness of the data. The ESC database is used to calibrate popular software estimating models like REVIC, PRICE-S, SEER-SEM, and COCOMO. Consequently, the parameters included in the database are similar to those included in the SMC database. While the SMC database includes 276 parameters, the ESC database has just over 60 parameters. Like the SMC's software database, many of the 60 parameters are not complete.

Data Scrub

Scrubbing the data is an important part of the methodology because not all the records in the SMC and ESC databases are useful. Each record must be checked for completeness. However, if all incomplete records were excluded from this analysis, there would only be a handful of records to perform the regression. Another cause for concern is that the records came from different sources and projects and each source or project emphasized different parameters. Still, there seems to be a common thread of relevant parameters that many of the records have. Consequently, when checking for completeness of records, only a subset of the fields will be required for inclusion in the analysis. This subset of fields is discussed more fully in the Data Selection section of this chapter.

Additionally, the data will be reviewed for potential errors or other discrepancies that could influence the outcome of the analysis. Those records that have errors or discrepancies will be excluded from the sample.

Data Selection

The key to the data selection is to find that relevant thread of important variables while at the same time striving for a large enough sample to perform the regression analysis. To increase the sample size, the number of parameters included in this review will be reduced from all possible parameters to only the most seemingly pertinent parameters. The parameters selected from the databases are based on basic attributes explained by Capers Jones in his book, *Estimating Software Costs,* and mentioned in Chapter II of this thesis. These parameters include the following:

1) Rate at which a project's requirements may change
2) Developing team's experience with this kind of project
3) The standards that will be employed, i.e. ISO, DoD
4) Programming languages utilized
5) Programming processes or methods
6) Reusable code
7) Development tools used
8) Office dynamics/environment
9) Schedule pressure (internal or external)
10) Complexity of the project

(Jones, 1998: 6)

The records that include the parameters that closely match the descriptions mentioned above are included in the analysis. Including only these key parameters, instead of all possible parameters, will increase the sample size; while ensuring that the most pertinent elements are included for consideration.

Data Preparation

There are two types of independent variables, quantitative and qualitative, included in regression analysis (McClave et al., 1998: 579). Quantitative variables are measured on a natural numerical scale. For example, the variable "lines of code" is considered quantitative. Lines of code may be any value from zero on up to however many lines of code are necessary to complete a project. On the other hand, qualitative variables are categorical in nature. Many of the parameters mentioned above, like inherent difficulty or personnel attributes, are reported as being high, nominal, or low. These parameters are clearly qualitative. To make these qualitative variables useful in regression analysis, the categorical data should be converted into indicator variables that can be used in computer statistical models. For example, two indicator variables should be created to account for the categories of high, nominal, and low. Figure 3.2 illustrates how the categorical data may be converted into indicator variables.

	x_1	x_2
Nominal (base)	0	0
Low	1	0
High	0	1

Figure 3.2. Categorization of Attribute Indicator Variables

When the category is nominal, both the x_1 and x_2 indicator variables are zero. When the category is low, the x_1 variable is one and x_2 is zero. When the category is high, the x_2 variable is one and x_1 is zero. With the categorical variables thus coded and the data thoroughly scrubbed, the data is ready for analysis.

Regression Analysis

Regression is a mathematical predictive tool used to predict a future response. It is based upon the correlation of the independent and dependent variables. Correlation is a statistical relationship between two sets of measures, or metrics, in which interval changes in one measure are accompanied by interval changes (not necessarily the same interval) in the other measure (Conte et al., 1986: 144). It should be remembered that the correlation may have nothing to do with cause and effect, but simply demonstrates a mathematical relationship between variables. Regression can be used to show potential cause and effect of variable, but can be verified through careful design of experiments. In other words, cause and effect may be verified through scientific experimentation; which isolates and studies the relationships of variables and their behaviors.

Probabilistic Models. It's possible to use multiple regression to determine a probabilistic model that can predict the relationship between the dependent variables and independent variables. Probabilistic models contain two components. The first component is the deterministic relationship between the independent and dependent variables. The next component is the random error due to random phenomena that occurs that can't be accounted for or explained by the model (McClave et al, 1998: 430-431). Probabilistic models that contain higher order terms, like x^2 or $ln(x)$, or contain more than one independent variable are called multiple regression models. The general form of multiple regression models is illustrated in equation 3.1.

$$y = \beta_i + \beta_i x_i + \beta_i x_i + \ldots + \beta_i x_i + \in \qquad (3.1)$$

where

y = predicted response
x_i = set of explanatory variables
β_i = Coefficients (unknowns)
$\in$ = error factor

The dependent variable y is a function of the i independent variables, $x_1, x_2, \ldots, x_i$. The value of the betas for each independent variable determines the level of contribution of the variables, β_0 is the y-intercept. The random error discussed above is represented by $\in$ (McClave et al., 1998: 500).

This project will use regression modeling to fit the collected data to a linear model. The data will be fitted to the model using the least squares approach. The least squares approach minimizes equation 3.2.

$$\text{SSE} = \Sigma(y - \hat{y})^2 \tag{3.2}$$

where:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_i x_i \tag{3.3}$$

(McClave et al, 1998:435-436)

The least squares approach will be used to analyze the data and create a model because it is unbiased, simple to use, and easy to understand. Through linear regression, a multitude of variables may be examined, and then reduced to the key variables that best predict the dependent variable (Conte et al., 1986: 280). Additionally, because the baseline form that is necessary for nonlinear model building isn't available, linear regression becomes the clear choice. Furthermore, the least squares approach is easy to understand it is easier to explain the relationships between the independent variables and the dependent variable.

There are various approaches to completing a least squares analysis. These approaches include forward-stepwise regression, backwards-stepwise regression, and a mixture of the two (both backwards and forwards). Of the three methods, there really is not an approach that will yield a superior model. The three methods may or may not yield the same result, but the selection of a method is dependent upon the personal preferences of the analyst (White, 1999). One benefit of backwards-stepwise regression is that it considers all interactions and higher order terms of variables. These interactions or higher order terms may not be examined if an analyst is using forward-stepwise regression because the lower order terms must first be selected in order for higher order or interactions of variables to be considered (White, 1999). For this reason, this analysis of the variables will employ backwards-stepwise regression in review of the data.

Modeling Assumptions. There are four basic assumptions concerning the random error, ϵ, that must be adhered to for the above form of probability distribution. The assumptions are as follows:

1. The mean of the probability distribution of ϵ is zero.
2. The variance of the probability distribution of ϵ is constant for all setting of the independent variable x.
3. The probability distribution of ϵ is normal.
4. The values of ϵ associated with any two observed values of y are independent.

(McClave et al, 1998:444)

These assumptions are required to determine the level of reliability and to develop hypothesis tests associated with the model.

When the final model is selected, the residuals ($\epsilon_i = y_i - \hat{y}_i$, $i = 1, \ldots, n$) may be tested to determine whether the assumptions are met. Creating an overlay plot of the residuals and examining the plots for sequential trends is a good method for testing the independence. If there are no apparent trends, the data is probably independent. Independence will also be tested using the Durbin-Watson test. According to Arthur Jensen, the Durbin-Watson statistic is used to test for the presence of first-order autocorrelation in the residuals of a regression equation. The test compares the residual for time period t with the residual from time period $t-1$ and develops a statistic that measures the significance of the correlation between these successive comparisons.

$$d = \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=1}^{n}(e^2)} \tag{3.4}$$

where

d = Durbin-Watson Statistic

$\in$ = residual

t = time period counter

The statistic is used to test for the presence of both positive and negative correlation in the residuals. The null hypothesis is that there is no significant correlation.

The Shapiro-Wilkes test will test for normality. Shapiro-Wilkes is a statistical test indicating the likelihood that a sample is drawn from a normal distribution. A Shapiro-Wilkes test statistic that yields a small value assuming the null hypothesis, that the sample is drawn from a normal distribution is true. Consequently, the variables are considered sufficiently normal if the p-value is greater than .05. Additionally, plotting the residuals against the quartiles of a normal distribution is another way of testing for normality.

Creating the "Full Model". The first step in creating the model is to examine all the variables and determine possible relationships with other variables. The independent variables should be plotted graphically against the dependent variable. Trends may be discovered through visually examining each graph of variables. The trends may appear in many different forms or shapes including linear, exponential, logarithmic, or other relationships. Another thing that should be considered when reviewing the trends is if there are any data outliers. Outliers may indicate the records don't belong to this particular data set or may indicate other trends that haven't been considered. Outliers will be a difficult to investigate in this analysis because of the way the data is masked in both the SMC and ESC databases.

The data should be examined to see if there is a high degree of correlation between independent variables. This correlation may point towards problems with multicollinearity. What this means is that multiple variables are explaining the same phenomena. If there are significant problems with multicollinearity, redundant variables should be dropped from the model.

The next step is to check the interactions between the variables. These interactions may be plotted against the dependent variable and trends may be revealed using the same process mentioned above.

From the above analysis a full model will be created that includes all possible interactions discovered from the above analysis. The following represents the outline of the full model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 \ldots\ldots + \beta_i X_i \qquad (3.5)$$

Variable and interactions will also be included in the full model and may be in the form of the following variables: $\beta_j lnX_j$, $\beta_k X_k^2$, $\beta_l X_m X_n$, or any other shape or interaction that is determined from plotting the dependent variable.

<u>Determining Usefulness of and Removing Individual Variables.</u> The next is to take the full model and reduce it, or eliminate variables that don't add to the explanation of the relationships of the variables. Variables will be systematically removed from the full model based on their relative contribution to the model and whether other variables explain the same relationship. This reduction will whittle away variables that appear to

add the least to the explanation of the model. The reduction of the variables will follow the following heuristics:

1. Remove a variable or variables that appear to contribute the least to the model based on their p-values and multicollinearity.
2. If the variable is included in an interaction, consider the interaction before considering the individual variable.
3. If the variable is included in a high-order term, consider the higher order term before the lower order term.

(White, 1999)

After each reduction of the model, the newly reduced model will be tested to determine if the reduced model is statistically as good as the model before the reduction.

The contribution of each variable or the usefulness of a variable will be determined through hypothesis testing. The purpose of the hypothesis testing is to determine whether the slope of a variable is something other than zero. Consequently, each β_i is examined using the following tests:

Two-Tailed Test	***One-tailed Test***	
H_o: $\beta_i = 0$	H_o: $\beta_i = 0$	(3.6, 3.7)
H_a: $\beta_i \neq 0$	H_a: $\beta_i < 0$ (or $\beta_i > 0$)	(3.8, 3.9)

(McClave et al., 1998: 451-452)

For the variable to be useful, β_i should not be equal to zero if it's at two-tailed test. In other words, for a useful variable, the null hypothesis will be rejected. The hypothesis may be tested by comparing a test statistic, t, to critical values of t, t_α, for a given significance level, α. The significance level, α, will be a commonly used software metrics α, which is .05. That is, there is a 5% chance our conclusion is false(Conte et al., 1986: 136). The test statistic, t, is illustrated below:

$$t = \frac{\hat{\beta}_1}{s / \sqrt{SS_{xx}}} \qquad (3.10)$$

(Where the standard deviation of beta hat is estimated by the portion of the equation in the denominator)

We can reach the same conclusion reached by the test statistic t by using the observed significance level (p-value) of the test from a computer printout. Computer statistical packages, like JMP or Excel, provide a p-value for each variable that may be used in this hypothesis testing. If the p-value is less than the desired α, the null hypothesis may be rejected and the variable may be considered useful (McClave et al., 1998: 452-453).

Model Selection and Reduction. After less useful variables are systematically removed from the full model based on their p-values and multicollinearity, the reduced model will be statistically tested to determine if the reduced model is statistically equivalent to the model before the reduction. The full model will be reduced until reductions are no longer statistically produce a model equivalent to the model before reduction (White, 1999).

To determine if the reduced model is statistically as good as the model before reduction, the F-test will be employed. The test statistic (T.S.) for the F-test is illustrated in equation 3.11.

$$T.S.:F = \frac{(SSE_r - SSE_f)/(\beta_f - \beta_r)}{SSE_f / df_f} \qquad (3.11)$$

where

SSE_r = Sum of squared errors for the reduced model

SSE_f = Sum of squared errors for the model before reduction

β_f = Number of β parameters for the model before reduction

β_r = Number of β parameters for the reduced model

df_f = Degrees of freedom for the full model before reduction

The test statistic, F, is compared to F_α. If F is less than F_α then the reduced model may be considered statistically equivalent to the model before reduction (White, 1999). F_α is based on $v_1 = (\beta_f - \beta_r)$ numerator degrees of freedom (*df*) and $v_2 = df_f$ denominator *df* (McClave et al., 1998: 600). Once again, the significance level, α, will be a commonly used software metrics α, which is .05. That is, there is a 5% chance our conclusion is false (Conte et al., 1986: 136).

Model Validation

Creating the model is just one step in the process of model building. It is also necessary to confirm that the model actually does what it is intended to do (predict the amount of effort necessary to develop BMC3 software). One popular measurement is the coefficient of multiple determination, R^2. (McClave et al., 1998) The R^2 will show the internal adequacy of the model.

Additionally, the model should be tested to see how accurately it predicts software development effort by using data other than the data used to build the model. To confirm the model's accuracy, actual historical data will test the accuracy of the model. The historical data may be data held out from the original data set, or new data from new projects. In this analysis it will be difficult to collect or create new data in a timely fashion, therefore, data will be held out from the original data set to test the model. The adequacy of the model will use the same tests used by previous AFIT theses mentioned in the literature review; which included the following measures:

- The *MMRE* is less than 0.25.
- The *RRMS* is less than 0.25.
- The Predicted Level *l* (or estimated effort is within 25% of actual effort at least 75% of the time.

<u>The Coefficient of Multiple Determination (R^2).</u> The measure R^2 is the coefficient of multiple determination. The coefficient of multiple determination measures the percentage of variance accounted for by the independent variables used in regression analysis. The measure R^2 has a range from zero to one and signifies the amount of variance that the model accounts for. (Conte et al., 1986: 168-171) For example, if $R^2 = .85$, then the model accounts for approximately 85% of the variance. However, the statistic R^2 should be used with caution because adding variables to the model may artificially increase R^2. Still, statistical software packages like JMP provide an adjusted R^2 that takes into account the phenomena mentioned above. Because of the

large number of variables included in this model, the adjusted R^2 will be the statistic used to evaluate the internal consistency of the model.

Mean Magnitude of Relative Error (MMRE). The *MMRE* is a measure that is concerned with how well actual values and values the model estimates relate to each other. The *MMRE* is the average absolute relative error. The equation for *MMRE* is illustrated in equation 3.12.

$$MMRE = \frac{1}{n}\sum_{i=1}^{n} MRE_i \tag{3.12}$$

where

$$MRE = \left|\frac{Actual - Estimate}{Actual}\right| \tag{3.13}$$

(Conte et al., 1986: 172)

Therefore, when the *MMRE* is small, the model on average produces a good set of predictions. As mentioned in the previous section, the prior theses efforts used $MMRE \leq$ 0.25 as an indicator of a model adequately predicting effort; 0.25 will also be used in this analysis.

Prediction at Level l (PRED(l). If k is the number of projects in a set of n projects whose $MRE \leq 1$, then prediction at level l is defined as:

$$PRED(l) = \frac{k}{n} \tag{3.14}$$

Conte gives the following example, if $PRED(0.25) = .83$, then 83% of the predicted values fall within 25% of their actual values(Conte et al. 1986: 173). The AFIT theses used Conte's suggestion that the acceptable criteria for an effort prediction model is *PRED(0.25)* greater than or equal to 0.75; this criteria will also be used in this analysis.

Relative Root Mean Squared Error (RRMS). The final measure the previous AFIT theses used was the *RRMS*. The *RRMS* is useful in that it enables models to be compared in terms of the mean value of the error minimized by the regression model. To calculate the *RRMS*, first the mean squared error (*MSE*) should be calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Actual_i - Estimate_i)^2 \tag{3.15}$$

Next, the root mean square error (*RMS*) may be calculated as follows:

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Actual_i - Estimate_i)^2} \tag{3.16}$$

Then the *RRMS* may be calculated using the two equation illustrated above. RRMS is defined by:

$$RRMS = \frac{RMS}{\frac{1}{n}\sum_{i=1}^{n} Actual_i} \tag{3.17}$$

(Conte et al. 1986:173-175)

Once again the AFIT theses used Conte's suggestion that the acceptable criteria is $RRMS \leq 0.25$, which will also be used in this thesis.

Conclusion

Through the use of multiple linear regression, a model will be created using ESC and SMC databases. The model will then be tested for usefulness using the techniques suggested by Conte, Dunsmore, and Shen in their book, *Software Engineering Metrics* *and Models*. This newly created model may be compared to previously calibrated popular models. This comparison will show whether the calibrated models or the created model will most accurately estimate BMC3 software development efforts.

IV. Data Analysis and Model Building

Overview

This chapter describes the procedures and results of the analysis of the data and model building efforts described in Chapter III. The outline used in this chapter will follow the same format as the outline mentioned in Chapter III and illustrated in Figure 4.1. The identification of potential data sources will not be discussed because it was already discussed thoroughly in Chapter III.

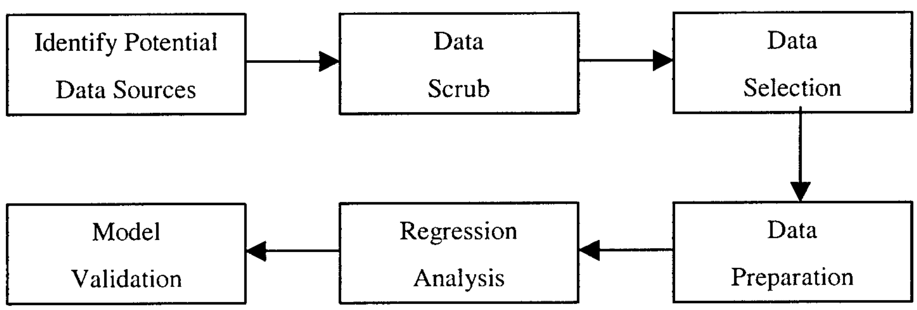


Figure 4.1. Methodology Process Flow Diagram

The data selection area will discuss why certain data points are chosen and why others are excluded. The data preparation section will discuss how the data was scrubbed and made useful for the purpose of regression analysis.

Then details of the regression analysis will be presented. The regression analysis will demonstrate why a specific model is selected as the best possible model based on the available historical data. The data will be analyzed in two ways. One way involves looking at many of the key variables as categorical variable; throughout this paper that model will be referred to as the categorical model. The other way involves looking at many of the key variables as continuous variable; throughout this paper that model will be referred to as the continuous model.

Finally, when the "best" model is selected, it will be tested for its adequacy based on the criteria mentioned in the previous chapter. The criteria include examining each model's internal and external validity. The internal validity is reviewed by examining the statistics created from the regression. The external validity is examined by applying Conte, Dunsmore, and Shen's criteria mentioned in the previous chapter.

Data Scrub

The historical data used in this study came from two sources. These two sources include the databases from the Electronic Systems Center (ESC) and the Space and Missile Systems Center (SMC). The SMC database includes about 2500 software development records. However, many of the records within this SMC database are incomplete. From the 2500 records only about seven of the records were useful for this analysis. On the other hand, the ESC, which contains only about 276 records, has much more useful data. From the ESC database, the remainder of the project's record (41 records) was collected.

Data Selection

The first step in selecting the applicable records includes isolating the records that are most like BMC3 software development efforts. Within these two databases, the records that are most like BMC3 efforts are the command and control software development efforts. Consequently, all the command and control records from both the databases were isolated.

Once the applicable records were selected, the goal in this regression analysis is to have the largest number of records possible to perform the analysis. The problem in this analysis is including as many records as possible while also including as many key parameters as possible. The key parameters included in this process were selected to represent the basic categories mentioned by Capers Jones in his book, *Estimating Software Costs.*

The key parameters include both the dependent and independent variables. The dependent variable is the total effort required to develop a software project. For this analysis, the total effort is measured in man months, or the amount of work one person can complete in one month. The independent variables include the following: effective size, actual new code, reused code, schedule months, application complexity, requirements volatility, timing constraint, personnel experience, programmers' language experience, specification level, resource allocation, modern practices experience, automated tool use, and programming language. Table 4.1 shows Jones' basic list of attributes and the variables from the databases that closely match or approximate the descriptions.

Table 4.1. Comparison of Jones' Basic Attributes and Included Variables

	Jones' Basic Attributes	Included Variables
1	Rate at which a project's requirements may change	Requirements volatility
2	Developing team's experience with this kind of project	Programmer experience, Programmer language experience
3	The standards that will be employed, i.e. ISO, DoD	Specification level
4	Programming languages utilized	Percentage of Assembly language
5	Programming processes or methods	Modern practices experience
6	Reusable code	Effective source lines of code, Reused source lines of code
7	Development tools used	Automated tool use
8	Office dynamics/environment	Resource location
9	Schedule pressure (internal or external)	Schedule in man months, Timing constraint
10	Complexity of the project	Application complexity,

Using these variables as selection criteria, 48 records were selected from the two databases, eight from the SMC database and 40 from the ESC database. However, if after further review one of the 48 records came under suspicion of validity. The record in question, reported that only 15 man months were required to develop over 15,000 lines of code. Based on an examination of the other records and tempered with reality, this record seemed highly improbable. Therefore this record was not included in the analysis.

The records mentioned above are included in Appendix A. For the sake of readability and format of this thesis, only the fields that were selected from the review mentioned above are included in the appendix. The complete records may be viewed in their entirety in either the SMC and ESC database.

Data Preparation.

The data was examined in two different ways. The reason two ways were used is that many of the independent variables are measures that are subjectively estimated by engineers working on a given project. Usually subjective measures like these are considered categorical variables in statistical analysis. On the other hand, the variables included in both of the databases mentioned above were derived from popular cost estimating models. In these models the variables are treated as continuous variables. For example, an estimator may determine a particular attribute has a categorical ranking of high, nominal, or low. Then the estimator inputs the variable in the model and the model uses that input to employ a numerical value based on a predetermined scale.

Within the databases the variables are reported with the following rankings: very low, low, nominal, high, very high, and extremely high. Consequently, because there has been two ways the variables have been examined in the past, both viewpoints will be examined in this analysis.

The first way the data was prepared was to take into account the categorical nature of the variables. Indicator variables were created to make the variables useful in the regression analysis.

Because the number of records was so small, using a large number of categorical variables is undesirable. This is undesirable because the greater the number of variables, the fewer degrees of freedom allotted to the model. A small number of degrees of freedom increases the area in the tails of the t-distribution and reduce a model's precision.

Consequently, this analysis grouped the categorical variables mentioned above into just three categories, which reduced the indicator variables per category from six to three categories. The various low ranking were grouped into one category, the various high rankings were grouped into another category, and the nominal rankings were simply reported as nominal. After the consolidation there are only three categories; which included low, nominal, and high. With the categorical variables grouped into the previous mentioned categories, the variables were then ready to code to make useful for the regression analysis. This coding involved creating indicator variables for each of the categories. An example of the coding is demonstrated in Figure 4.2.

	x_1	x_2
Nominal (base)	0	0
Low	1	0
High	0	1

Figure 4. 2. Illustration of indicator variables for categorical variables

The second way the variables were examined included taking advantage of the comparable format of the variables collected in the two databases. Originally data or the variables collected were in comparable formats to many of today's popular models. Both of the databases included variables that are traditionally used in the COCOMO model and other models that use similar variables employed in the COCOMO model.

The COCOMO model variables that are ranked high, nominal, and low have corresponding numerical values that are based on scales for each the variables. Consequently, this study also explored the relationship of the subjective measures mentioned above in a continuous fashion. Therefore, this study included converting the subjective variables as reported into numerical values based on the values provided by the COCOMO model. Figure 4.3 illustrates the values the COCOMO model uses for each of the variables that were included in this study.

Variable

	Application Complexity	Requirement Volatility	Timing Constraints	Personnel Experience	Language Experience	Specification Level
very low	0.70	n/a	1.23	1.29	1.14	0.70
low	0.85	0.91	1.08	1.13	1.07	0.85
nominal	1.00	1.00	1.00	1.00	1.00	1.00
high	1.15	1.19	1.04	0.91	0.95	1.15
very high	1.30	1.38	1.10	0.82	n/a	1.30
extra high	1.65	1.62	n/a	n/a	n/a	1.65

	Resource Support Location	Modern Practices Experience (based on SLOC) 2k SLOC	8k SLOC	32k SLOC	128k SLOC	512k SLOC	Automated Tool Support
very low	n/a	1.25	1.30	1.35	1.40	1.45	1.24
low	n/a	1.12	1.14	1.16	1.18	1.20	1.10
nominal	1.00	1.00	1.00	1.00	1.00	1.00	1.00
high	1.12	0.90	0.88	0.86	0.85	0.84	0.91
very high	1.23	0.81	0.77	0.74	0.72	0.70	0.83
extra high	1.35	n/a	n/a	n/a	n/a	n/a	n/a

(Boehm, 1981)

Figure 4.3. COCOMO Rating For Descriptive Variables

Another variable that was considered during the preparation of data phase was the programming language variable. Based on the study mentioned in Chapter II that was conducted by the Naval Cost Analysis Agency, there is not a significant difference between different third-generation languages. However there is a significant difference between second and third generation languages. Consequently, the presentation of the data needed to reflect the different percentages of second and third generation languages. Many of the projects included a variety of different languages. Still, assembly language is the language that seemed to make a difference in the amount of effort necessary to produce lines of code. Therefore, for the purpose of this regression analysis the language variable was expressed as a percentage of the project coded in assembly language. The language variable was included in both versions of the data set, which includes the variables presented in the categorical fashion and the continuous fashion.

Regression Analysis

Once the data was selected and properly prepared it was ready for the regression analysis. The regression analysis was performed in a computerized statistical package called JMP IN©. The data was looked at a number of ways, which included various combinations and mathematical variations of the data. The various combinations and mathematical variations of the data were based on observations of the data. The data observations were derived from completing scatter plots of many of the plausible combinations of the variables. For example, one of the variables that was included in the final model was the variable *AC_H*. The variable *AC_H* was plotted with the effective size, as illustrated in Figure 4.4. The scatter plot revealed a plausible trend line equation with an associated power. Then this variable is transformed with the power observed in the scatter plot and included in the full model. All scatter plots' that were considered in this analysis are included in Appendix B.

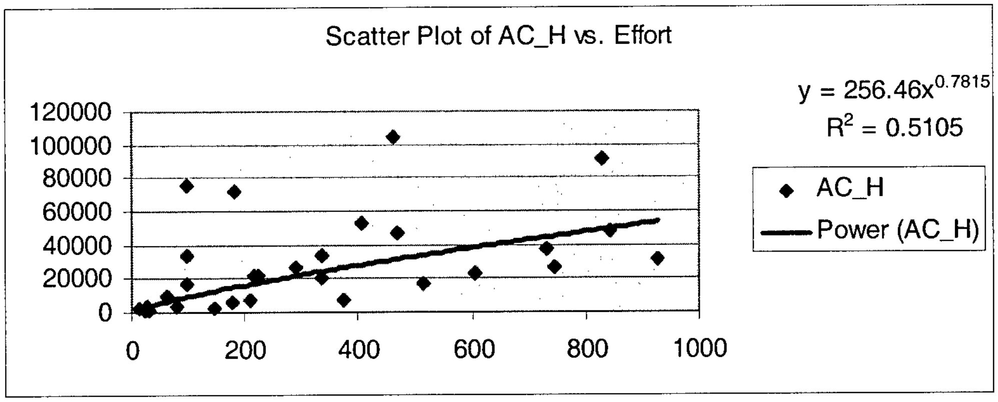


Figure 4.4 Scatter Plot of AC_H vs. Effort

One relationship that seems important to include in the analysis is an interaction between the descriptive categorical variables and the size of the project. Intuitively it makes sense that something like the complexity of a project or any of the other categorical variables would have a greater effect the larger (or smaller) a project became. Furthermore, when these interactive relationships were plotted in relationship to effort, the interactive relationship was confirmed with many of the categorical variables.

Within the graphs in Appendix B, trend lines demonstrate the most perceptible relationships between the independent variables and the dependent variable. The trends that appeared to be the strongest were included in the full models that are illustrated in Tables 4.2 and 4.3. In the categorical model, Table 4.2, only the interactions that showed a higher level of correlation were included in the full model. These relationships included the scatter plots that demonstrated a trend line with an R^2 greater than 0.5. Additionally, the scatter plots that only had a few number of records (like *PE_H*, *RSL_H*, *AC_H*, and *RV_H*) were not included as interactions in the full model because there were not enough records to reliably establish a trend. The interactions that were included in the full models demonstrated in their scatter plots that there was some degree of diminishing returns. These diminishing returns are expressed as the power that the variables are raised to in the model as demonstrated in the trend lines of the scatter plots.

Table 4.2. Definition Of Variables Included In The Categorical Model

Term	**Definition**
Norm_siz	Effective source lines of code which include both new and reused lines of code
SLOC_N	New source lines of code
SLOC_R	Reused source lines of code
Sched	Schedule in man months
AC_H	High application complexity
RV_L	Low requirements volatility
RV_H	High requirements volatility
TC_H	High timing constraint
PE_L	Low programmer experience
PE_H	High programmer experience
LE_L	Low programmers' language experience
LE_H	High programmers' language experience
SL_H	High specification level
RSL_H	High resource location
MPE_L	Low modern practices experience
MPE_H	High modern practices experience
AT_L	Low automated tool use
AT_H	High automated tool use
language	Percentage of assembly language
size^0.75	$(\text{effective size})^{0.75}$
Achp	$((\text{high application complexity}) \times (\text{size}))^{0.78}$
Rvlp	$((\text{high timing constraint}) \times (\text{size}))^{0.89}$
Tchp	$((\text{low programmers' experience}) \times (\text{size}))^{0.76}$
Pelp	$((\text{low language experience}) \times (\text{size}))^{0.65}$
Lelp	$((\text{low tool use}) \times (\text{size}))^{0.73}$
Atlp	$((\text{high tool use}) \times (\text{size}))^{0.80}$
Athp	$((\text{low requirements volatility}) \times (\text{size}))^{0.80}$
*AC_H*Norm_siz*	(high application complexity) x (size)
*TC_H*Norm_siz*	(high timing constraint) x (size)
*PE_L*Norm_siz*	(low programmers' experience) x (size)
*LE_L*Norm_siz*	(low language experience) x (size)
*AT_L*Norm_siz*	(low tool use) x (size)
*AT_H*Norm_siz*	(high tool use) x (size)
*RV_L*Norm_siz*	(low requirements volatility) x (size)

Table 4.3. Definition Of Variables Included In The Continuous Model

Term	**Definition**
language	% Assembly language
Size	Source lines of code
SLOC N	New source lines of code
SLOC R	Reused source lines of code
schedule	Number of man months
AC	Application complexity
RV	Requirements volatility
TC	Timing constraint
PE	Programmers' experience
LE	Language experience
SL	Specification level
RSL	Resource location
MPE	Modern practices used
ATS	Automated tool use
size ac	(Application complexity * size)$^{0.69}$
size rv	(Requirements volatility * size)$^{0.78}$
size tc	(Timing constraint * size)$^{0.74}$
size pe	(Programmers' experience * size)$^{0.70}$
size le	(Language experience * size)$^{0.72}$
size sl	(Specification level * size)$^{0.76}$
size rsl	(Resource location * size)$^{0.74}$
size mpe	(Modern practices experience * size)$^{0.69}$
size ats	(Automated tool use * size)$^{0.74}$
size/sched	(Size/schedule)
$ss^{0.75}$	(Size/schedule)$^{0.75}$
$Size^{0.75}$	(source lines of code)
*AC*Size*	Application complexity * size
*RV*Size*	Requirements volatility * size
*TC*Size*	Timing constraint * size
*PE*Size*	Programmers' experience * size
*LE*Size*	Language experience * size
*SL*Size*	Specification level * size
*RSL*Size*	Resource location * size
*MPE*Size*	Modern practices experience * size
*ATS*Size*	Automated tool use * size

Like the categorical model, the continuous model's scatter plots were examined to determine plausible relationships between the dependent variable and the interactions of the independent variables. Once again, the scatter plots demonstrated that there are probably, to some degree, diminishing returns in the relationships.

Additionally, the continuous model's diminishing returns are expressed in terms of the power of the variables. These powers were derived from the best-fitted trend line for each of the scatter plots. The variables powers are illustrated in the equations of the trend lines in the scatter plots and then incorporated into the full model, which is illustrated in Table 4.3.

Through a review of those scatter plots, the most significant relationships were determined and placed in the model. Scatter plot graphs were completed for both versions of the data set. Then and the most plausible relationships of the variables were included in the full model in preparation of the regression analysis. Table 4.2 and 4.3 lists all the variables included in the full or unreduced model. Also included in the tables are the definitions for each of the variables. The two lists of definitions are similar in that they both come from the same databases. Still, there are differences. The main difference is that in Table 4.2 many of the explanatory variables are categorical variables and are designated as being either high or low (nominal is considered the baseline). On the other hand, the variables in Table 4.3 are continuous in nature. Additionally, when the variables were examined as either categorical or continuous variables there appeared to be differences in the importance of variables and how those variables may be combined. The relationships of these variables were apparent from viewing their scatter plots.

The full model for both the categorical model and the continuous model were built using the basic regression model format technique presented in chapter III. Each variable mentioned in Table 4.2 or 4.3 represents one of the x_i variables in the model. The coefficient, β_i, is determined was using the least squares method in JMP IN©.

Selection of the Final Model. Once the full model was constructed, the analysis to select the final model was initiated. Appendix C details the analysis for the selection of the categorical variable model and Appendix D details the selection process for the continuous variable model. As illustrated in Appendices C and D, the less useful variables were systematically removed from the full model based on their p-values and possible multicollinearity. Using the heuristics mentioned in Chapter III, the largest p-values were removed first until no p-value over 0.05 remained. In the appendices, the full model is illustrated; then the variable that appears to add the least to the model is indicated as a candidate for removal. The candidate variable is removed from the model then the new statistics for the model with reduction is examined. With the newly reduced model's statistics, the model is tested using the F-test mentioned in Chapter III. The F-test is used to determine if the model is statistically equivalent to the model before reduction. Then the process continues until reductions can no longer be made to the model without affecting its equivalency.

The model with the categorical variables (or the categorical model) went through 23 iterations of reductions before there could be no further reductions. The first reduction was selected based on the Chapter III's heuristics and the information derived from the correlation matrix, which is included in Appendix E. The relationships in Table 4.4 cause concern because of their values reported in the correlation matrix.

Table 4.4. List Of Problematic Terms From The Correlation Matrix

Variables:			
Norm_siz	and	*SLOC_N*	0.8849
Norm_siz	and	$size^{0.75}$	0.9907
$size^{0.75}$	and	*SLOC_N*	0.8753
LE_H	and	*LE_L*	0.8607

Norm_siz and *SLOC_N* have a large degree of correlation because the *SLOC_N* (new lines of code) is a function of normalized size (*Norm_siz*). The high degree of correlation between the size variables and $size^{0.75}$ should be expected because $size^{0.75}$ is the higher order term of the normalized size. *LE_H* and *LE_L* also cause some concern and one or the other variables should be removed during the normal course of reductions.

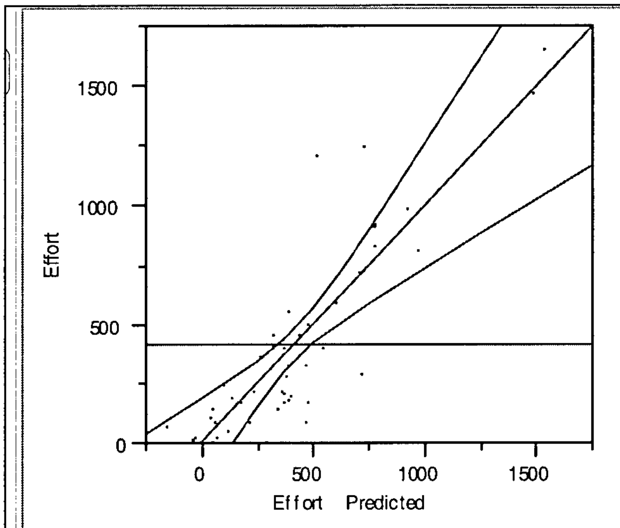
The reductions outlined in Appendix C start with the removal of two variables, *SLOC_N* and *SLOC_R*. *SLOC_N* and *SLOC_R* were removed first for two reasons. First, they had high p-values. Second, There is a high degree of correlation between *SLOC_N* and *Norm_siz*, therefore, one of these two variables should be removed. Normal size captures both the information of *SLOC_N* and *SLOC_R*, consequently, both *SLOC_N* and *SLOC_R* were removed. The remainder of the reductions strictly followed the heuristics mentioned in Chapter III. The variable with the highest p-value was removed provided that it isn't a lower order term or one of the terms included in an interaction.

The model with the continuous variables (or continuous model) went through 18 iterations of reductions before its reductions were no longer statistically equivalent. The continuous model reductions are illustrated in Appendix D. *SLOC_N* and *SLOC_R* were the first two variables removed from the continuous model for the same reason mentioned above for the categorical model. The correlation matrix that is included in appendix E revealed that the size variables, *Norm_siz* and *SLOC_N*, had a high degree of correlation with all the interaction terms. In the table in Appendix E, these relationships are highlighted. It makes sense that there would be a high degree of correlation because these interaction variables include size as a multiplier. Additionally, Appendix E reveals that there is a high degree of correlation between the interactive terms and other interactive terms. This too makes sense because all the interactive terms contain the same variable, *Norm_siz*. This correlation will have to be tolerated if the interactive variables are to be included as part of the model. Like the categorical model, the continuous model followed the heuristics mentioned in Chapter III for each of the reductions.

The significance level used in both model reduction analyses was 0.05 as suggested in Chapter III. Through this process the simplest model and hopefully the easiest to explain model was produced.

<u>Results of the Reduced Categorical Model.</u> The reduction of the categorical model produced a simplified model that included ten separate variables. Of these variables, there were two continuous variable, effective size and schedule, and four separate categorical variables. From the four categorical variables four interactions were derived. The final reduced model is illustrated below in equation 4.1. The graphical results of the least squares line for the equations are presented in Figure 4.3. The line in Figure 4.3 illustrates the calculated least squares line and points are the plots of the actual and predicted values of each record used to create the model.

$$
\begin{aligned}
Effort = & -0.000435(Norm\_siz) + 4.0452889(Sched) - 457.0239(AC\_H) \\
& -17.96349(RV\_L) - 262.6671(TC\_H) - 321.2767(AT\_L) \\
& + 0.5892851(achp) - 0.026304(AC\_H*Norm\_siz) \\
& + 0.0137056(TC\_H*Norm\_siz) - 0.007164(RV\_L*Norm\_siz) \\
& +296.21152
\end{aligned} \tag{4.1}
$$



Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	10	5503146.5	550315	11.3994
Error	36	1737927.9	48276	Prob>F
C Total	46	7241074.4		<.0001

Figure 4.3. Graph of Least Squares Line of Categorical Model

The model has an R^2 of 0.7599 and an adjusted R^2 of 0.6933, which accounts for the degrees of freedom within the model. In other words, this model accounts for about 69% of the variance in the least squared regression.

Two of the key assumptions that must be met to use the least squares method go hand-in-hand; these being the random error's mean of the probability is zero and the probability distribution is normal. Normality was tested by examining the distribution of y's for the residuals of the model; Figure 4.4 is a graph of those residuals. At first glance, the graph appears to have some problems with normality. However, there are two

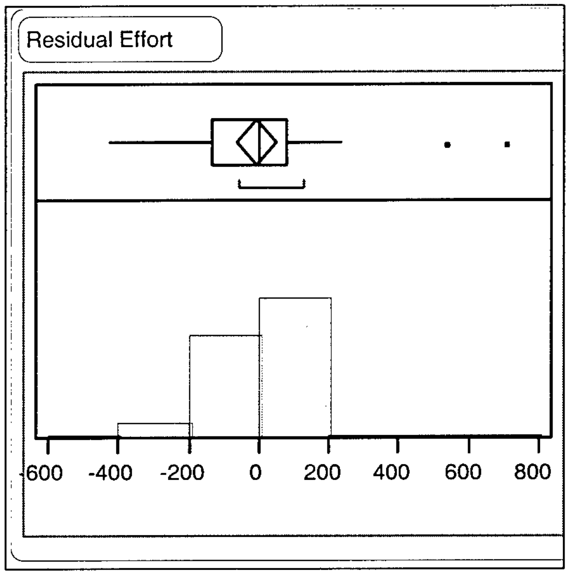


Figure 4.4 Histogram of Categorical model residuals

significant outliers. Those outliers are records 28 and 38 from the data listed in Appendix A. These two outliers disguise the shape of the graph and, in effect, conceal a plausible distribution of the residuals. Because the two databases that these records are extracted from are masked, it's difficult to verify these two records. Therefore, for considering the plausible normality of the residuals, the residuals are considered without these outliers. When the graph without the two outliers is examined, the residuals appear to be much more normally distributed. Figure 4.5 illustrates the residuals without the outliers. Additionally, the Shapiro-Wilkes test mentioned in Chapter III is a good indicator whether or not residual distributions are sufficiently normal. Chapter III mentioned that if the p-value of the test is greater than 0.05 the residuals are sufficiently normal. The p-value for this distribution is 0.163; therefore, these residuals are considered normally distributed for this purpose. Because this test is robust, it may be assumed the residuals are normal.

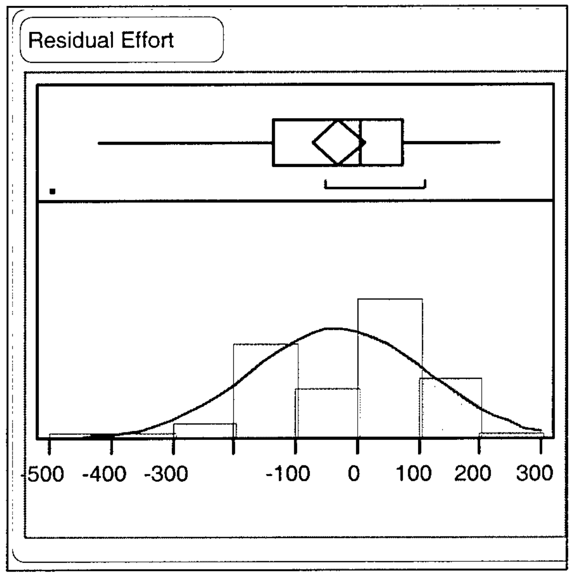
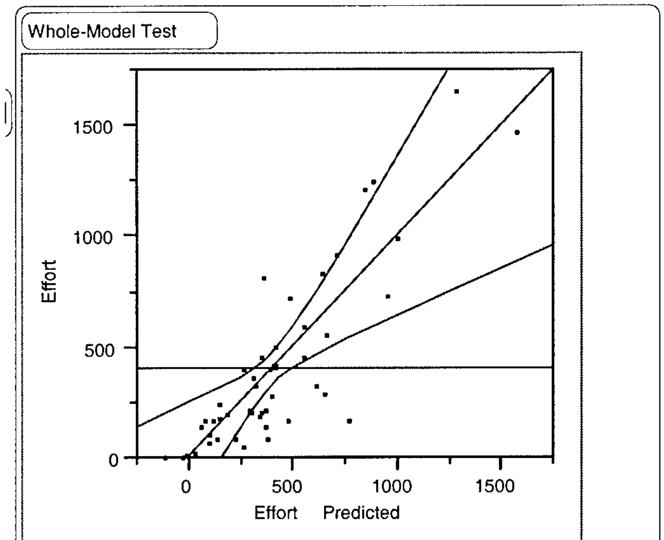


Figure 4.5. Histogram of Categorical model residuals (less outliers)

The independence of the variables was tested using the Durbin-Watson test. The test resulted in a test statistic of 2.07. Further investigation of the independence of the variables is difficult to conduct because of the masked nature of the data. Therefore, to be able to complete the regression analysis, independence of the variables is assumed.

Results of the Reduced Continuous Model. Like the categorical model, the reduction of the continuous model produced a simplified model, which included 16 separate variables. Similar to the previous model, this model includes effective size and schedule as variables in addition to seven other descriptive variables. Through a review of the interactions and observations of their scatter plots in relation to effort, there seems to be an interaction between size and schedule.



Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	16	5533480.6	345843	5.7379
Error	31	1868489.8	60274	Prob>F
C Total	47	7401970.4		<.0001

Figure 4.6. Graph of Least Squares Line of Continuous Model

In addition to an interaction between size and schedule, six other interactions were observed between the descriptive variables and effective size. The final reduced model is presented in equation 4.2 and the equation's results are illustrated in figure 4.6. The line in figure 4.6 illustrates the calculated least squares line and the points are the plots of the actual and predicted values of each record used to create the model.

$$\begin{aligned} Effort = {} & -0.035862(Size) - 9.654109(schedule) - 1017.302(AC) + 494.62747(RV) \\ & + 9497.2342(TC) + 2382.1704(LE) - 1341.63(SL) - 10095.92(RSL) \\ & - 1543.876(MPE) - 23.82223(size\ tc) + 6.1262998(size\ sl) \\ & + 17.295797(size\ rsl) - 0.32946(size/sched) + 0.8773182(TC*Size) \\ & - 0.277077(SL*Size) - 0.607228(RSL*Size) + 2006.7501 \end{aligned} \tag{4.2}$$

The same methods were used in the continuous model as the categorical model for testing the key assumption, which include testing if the random error's mean of the probability is zero and if the probability distribution is normal. Figure 4.7 is a graphical representation of the continuous model's residuals. The Shapiro-Wilkes test for normality returns a p-value of 0.6615. Based on the visual review of the model's residual graph and the Shapiro-Wilkes test, the assumption that random error's mean of the probability is zero and the probability distribution is normal is upheld.

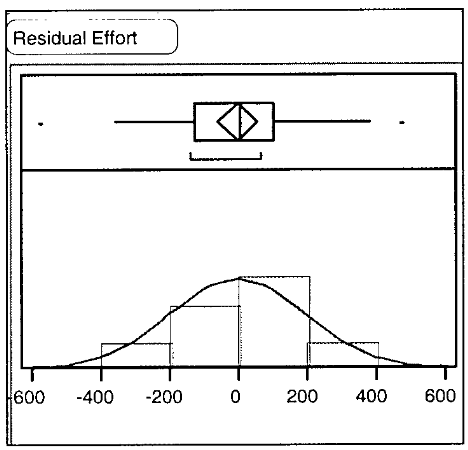


Figure 4.7. Histogram of Continuous model residuals

Like the categorical model, the independence of the variables was tested using the Durbin-Watson test. The test resulted in a test statistic of 1.59. Because the continuous model's data is basically the same data as the categorical model's data, further investigation of the independence of the variables is difficult to conduct because of the masked nature of the data. Therefore, like the categorical model, the independence of the variables is assumed.

Model Validation

Model Adequacy. There are two parts to model evaluation, which include determining the internal and external adequacy of the models. First, as discussed in the previous section, the R^2 and the adjusted R^2 demonstrated the adequacy of the models. Second, the model evaluation should determine how well the model might predict effort for projects other than projects used to develop the model. This is accomplished by using data other than the data used to create the model, where both the independent and dependent variables are known.

The data used to test the models were extracted from the same databases that the models were created from. This was possible because of the way the data used to build the models were originally selected. The records that were initially extracted were only the records that contained reported information in all the fields used to build the full models. Because only a portion of the actual number of records from the two databases was used to create the models, there were still records available to test each model. This is possible because when the models were reduced and had a smaller number of variables more records fit the criteria necessary to be included in the data set.

The final equation for the categorical model (Equation 4.1) requires six fields of independent variable, plus their interactions, to predict the dependent variable, effort. After searching through the databases, 14 records were found that met the criteria for Equation 4.1.

Table 4.5. Tested records with actual and estimate effort for Categorical model

ID	Sched	Size	AC_H	RV_L	TC_H	AT_L	Actual Effort	Estimated Effort
23-1	41	216,088	1	0	1	0	5,007	3,604
23-2	41	60,475	0	0	0	0	597	436
42-1	54	8,127	0	0	1	1	120	39
44-1	48	54,192	1	1	0	0	1,219	512
50-1	69	40,702	1	1	1	0	334	893
50-2	44	6,450	0	1	0	0	461	407
50-5	63	26,933	1	1	1	0	299	641
50-8	66	14,809	1	1	1	0	1,414	397
50-9	66	14,817	1	1	1	0	937	397
50-10	45	10,991	0	1	0	0	851	377
50-11	56	25,212	1	1	0	0	300	498
50-12	65	12,774	1	1	1	0	176	342
50-13	67	21,502	1	1	1	0	213	551
50-15	69	32,100	1	1	0	0	1,011	580

Model Validity. Table 4.5 lists the 14 records and the values that were reported in the records. The independent values for each of the records were inputted into Equation 4.1. The result of each equation is listed in the final column, estimated effort. The estimated effort was compared to the actual effort for each of these records. The estimated and actual results were evaluated using the measures suggested by Conte, Dunmore, and Shen (Conte et al., 1986: 168-171). These results are reported in Table 4.6.

Table 4.6. Model Validity Measures

	Categorical Variable Model	Continuous Variable Model	Suggested Standard
MMRE	0.73	1.45	< 0.25
RRMS	0.64	1.06	< 0.25
pred (.25)	7%	0%	> 75%

There was also an attempt to validate the model by using the same test set and inputting them into popular models used by the DoD. However, these popular models require many more input variable than are reported in the records used for validation. Appendix F illustrates the required fields for five popular models used by the department of defense. The matrix in Appendix F lists possible input variables and then indicates which of those variables are required for inputs to that particular model. Also illustrated in Appendix F are the reported categories for each of the records used in the above validation. The fields that are reported by a particular record are shaded to indicate a response. The conclusion of this matrix provided in Appendix F is that none of the records used to validate the model mentioned above have enough complete fields to adequately test the popular models listed in the appendix.

The continuous variable model was tested in the same way as the categorical variable model. The final equations for the continuous model (Equation 4.2) requires nine fields of independent variable, plus seven interactions, to predict the dependent variable, effort. After searching through the databases, six records were found that met the criteria for Equation 4.2. Table 4.7 lists the six records and their reported values for the pertinent fields. The independent values for each of the records were inputted into Equation 4.2. The result of each equation is listed in the final column, estimated effort. The estimated effort was evaluated in the same way and with the same measures as the categorical model. The resulting statistical measures are displayed in Table 4.7.

Table 4.7. Tested records with actual and estimate effort for Categorical model

ID	Size	AC	LE	MPE	RSL	SL	TC	RV	schedule	Actual Effort	Estimated Effort
16-0	20,075	1.30	0.90	0.86	1.00	1.00	1.00	0.91	24	125	82
18-1	35,000	1.65	0.95	1.35	1.23	1.65	1.04	1.62	30	262	(176)
18-2	22,400	1.65	0.95	1.35	1.23	1.65	1.10	1.62	32	152	(383)
18-3	39,500	1.65	0.95	1.35	1.23	1.65	1.10	1.62	32	301	(151)
42-1	8,127	1.00	0.95	0.88	1.00	1.00	1.04	1.00	54	120	278
44-1	54,192	1.30	0.85	0.74	1.00	1.30	1.00	0.82	48	1,219	801

V. Results and Recommendations

Overview

The purpose of this thesis was to develop a model that could accurately estimate the development costs for command and control software systems. As mentioned earlier in this thesis, the Department of Defense simply does not have an adequate method for estimating these development costs. Many efforts in the past have revolved around using existing Department of Defense data to calibrate popular software estimating models. Still, no attempt within the Air Force Institute of Technology has been made to use the Department of Defense data to create a model based on the in-house data. Consequently, the efforts described in this thesis use the in-house data to develop a model specifically for the purpose of estimating Department of Defense command and control software systems.

The records that were selected from both the ESC and SMC databases were analyzed in two different ways, as mentioned in chapter 4. The descriptive variables in both the databases were analyzed as either categorical or continuous variables. These two ways of analyzing the variables, led to creating two different models. The results from the two different models varied. One of the models or regressions emerged as the more useful model, or appeared to be statistically better and made more sense. The next few sections will discuss the results of both models and why a one model appears to be better than the other.

Results of the Two Models

Based on the measurements proposed by Conte, Dunsmore, and Shen; it seems to be clear which of the two models is better than the other. As shown in Table 5.1, the categorical variable model is closer to meeting the objective criteria than the continuous variable model. Still, it is difficult to compare the two models against each other based on these criteria because the data set used to measure the criteria was different for each version of the model. The difference occurred because of the different variables that were required for each version of the final models. Records were selected from the original databases if they had all the variables required by a particular model. The resulting subsets of records for each of the models contained very little overlap (only two records.)

Table 5.1. Results for Categorical and Continuous Variable Models

	Categorical Variable Model	Continuous Variable Model	Suggested Standard
MMRE	0.73	1.45	< 0.25
RRMS	0.64	1.06	< 0.25
pred (.25)	7%	0%	> 75%

Table 5.2. Results for commonly used models

Author (Year)	Cost Model	MMRE	RRMS	Pred (0.25)	MMRE	RRMS	Pred (0.25)
Kressin (95)	SLIM	0.62	n/r	0.00	0.67	n/r	0.00
Rathmann (95)	SEER-SEM	0.53	1.03	0.31	0.31	0.30	0.29
Mertes (96)	CHECKPOINT	0.19	0.15	0.50	0.17	0.16	0.50
Marzo (97)	SAGE (SMC)	0.40	0.59	0.37	0.35	0.56	0.41
Marzo (97)	SAGE (ESC)	0.38	0.68	0.27	0.37	0.53	0.22

For the same reasons as mentioned above, it was impossible to compare the results of the models developed from this analysis and the results of popularly used software estimating models. Prior AFIT theses efforts used the same criteria mentioned above to test the "goodness of their models. However, each of these efforts used a different data set to test the criteria. Therefore, it is difficult to compare the models to each other because of the differing data sets used to test each model. The results of the popular model's tests are shown in Table 5.2.

Another way to look at the two models is to examine the outputs of the models. The continuous variable model had some serious problems in estimating the amount of effort necessary to develop software. For example, many of the estimated results were negative numbers. Intuitively it doesn't make a lot of sense to have negative effort to complete a software project.

Additionally, when the model itself is examined, it becomes apparent why the model yields these atypical negative numbers. Some of the coefficients within the model simply don't make sense. For example, the coefficients that determine the effect of schedule months, resource location, and specification level are all negative. In other words, as these variables increase, the amount of effort necessary to complete a software project decreases. Intuitively, this cannot be true; as these variables increase, the amount of effort necessary should also increase. These problems could possibly arise from the multicollinearity among the variables in this model. As demonstrated in the correlation matrix in appendix E, many of the variables in this model have probable multicollinearity due to the fact that the problematic interactive variables have very similar inputs.

Conversely, the categorical variable model's coefficients for each of the variables seem to make sense intuitively. The only variable that had a questionable coefficient is the size variable. However, the marginal contribution of this variable is minimal and the true effects of size are probably captured in the terms that include size as an interactive variable. Additionally, the VIF as illustrated in Reduction 23 in Appendix C for the size variable is high, which indicates a potential problem with multicollinearity. The variable probably has high multicollinearity because it is included as interactive terms. Consequently, the size variable may be acting as a correction factor for the other size variables included in interactions. Still, this variable must be included because it is included in other interactive terms within the model.

All the other variables' coefficients seem to act in ways that they are expected. Some of the variable have interactive and higher order terms and should be considered as a group or a family of terms. For example, the marginal influence that *AC_H*, *AC_H*Norm_siz*, and *achp* should be considered together. When all the variables of this family of variables are considered together, the marginal effect of these variables is that when application complexity for program with an effective size greater than about 13,500 SLOC is high, the amount of effort increases as the ESLOC increases as illustrated in figure 5.1.

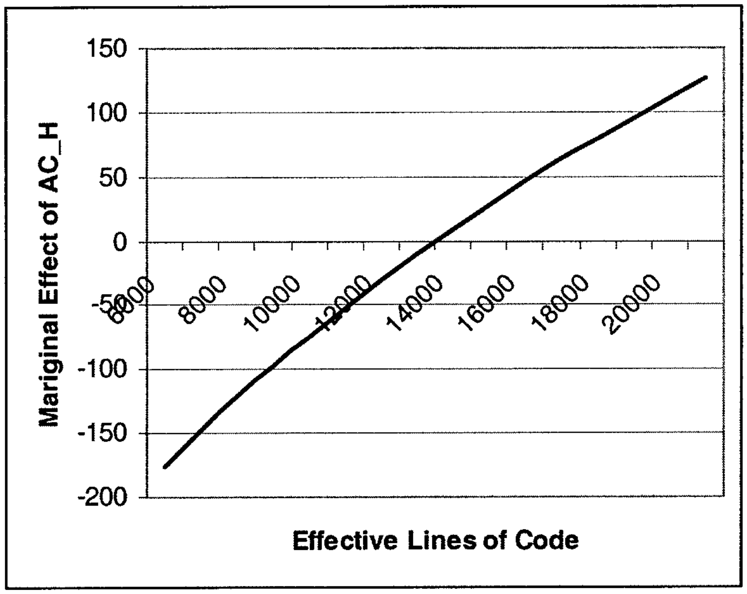


Figure 5.1. Marginal effect of *AC_H*

Like the variable *AC_H*, the timing constraint variable (*TC_H*) also must be considered as a group to get the full picture of what the marginal effect of a high timing constraint. The point at which this family of variables crosses from negative to positive is at about 2,000 ESLOC as illustrated in figure 5.2.

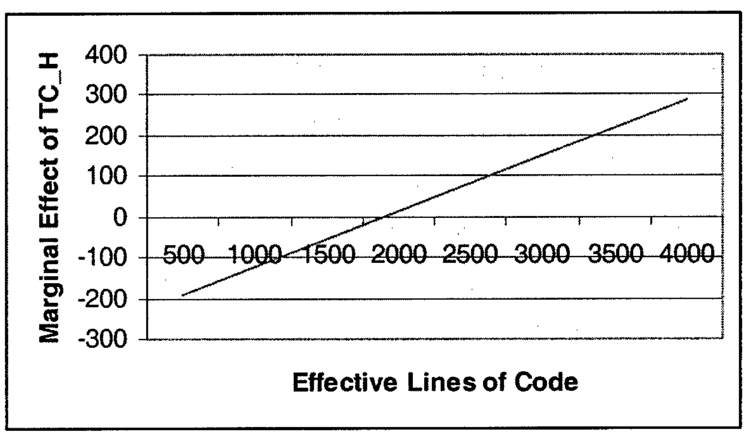


Figure 5.2. Marginal effect of TC_H

Still, the metrics that are illustrated in table 5.1 don't necessarily provide a lot of confidence for this model. Furthermore, the same metrics that were performed on the popular models don't necessarily give a lot of credence to those models either. The poor results of the tests for all the models may be that they are poor models, or it may be that the records used to test the models aren't representative of the true population of software development efforts. This is really the problem of trying to predict future performance or effort based on past efforts; which models are most applicable or will yield the best answer?

Different models have different strengths and weaknesses. The preferred model in this thesis (the categorical variable model) has a strength of requiring only a small number of input variables for the model. Furthermore, it is easy to use and easy to understand. Still, probably the best thing to do is use a combination of models to estimate future performance. Boehm states that it is important to us a combination of techniques in order to avoid the weaknesses of any single method and to capitalize of their joint strengths. (Boehm, 1981: 323)

One model may be used to develop an estimate, and another model may be used as a crosscheck for that estimate. Can anyone be 100% confident their estimate is correct? Of course not, but through a use of crosschecking models, one may be confident that an estimate is somewhere in the ballpark. Additionally, as differences arise in estimates, one can investigate why there is a difference and what it is composed of and try to determine which model is more accurately estimating reality.

Still, the model presented in this thesis should be used with caution. It seems to provide anomalous results when it is in either extreme of its estimating range. In other words, when ESLOC is extremely high or low this may not yield good results.

Limitations

The following items are limitations to the methodology used in this thesis and should be considered when evaluating the results of the analysis.

1) One of the biggest problems with the data provided by ESC and SMC is that there is practically no way to confirm the data. There may be errors or other types of irregularities within the data, but because the projects and the contractors are masked from the database there can be no confirmation. Additionally, many different people collected the data over many different years. This could cause a lot of discrepancies, which may include how code is counted or what a particular person determines is a high, nominal, or low ranking. When collecting data from various sources, consistency is paramount; still, there is no way to confirm the consistency of the data used in this analysis.

2) Another limitation with this analysis is that many people within the industry feel that linear models cannot accurately predict the complex relationships involved in software development efforts (Conte, 1986: 280). Nonlinear models are difficult to build and explain the relationships once they are built. On the other hand linear models are simpler to build and easier to explain. Therefore, through the use of higher order terms, this model will attempt to model the complex software development relationships.

3) Another concern is the applicability of data to new projects being estimated. As technology increases, programming languages and techniques will surely change. The question is whether or not the historical data from these databases can be used to estimate future software development efforts. When an estimator uses regression models they need to make sure the new project being estimated is comparable to the projects the model was created from. If the attributes of the new development effort are significantly different from the historical data, the regression model probably shouldn't be used.

Recommendations

Probably the biggest single obstacle this thesis effort ran into was the lack of data, or the lack of quality data. Both the ESC and SMC databases contain numerous records that described historical results of command and control software development efforts. Still, neither of these databases is very thorough in completing all the fields within the records of the databases. Additionally, the data that comprise both of these databases were collected over a long period of time, by different people, and probably for different purposes. Because of this, the consistency within the databases as far as definitions of the variables is suspect. Consequently, one of the greatest things that could be done in this area is improve the collection of data for software development efforts.

Another thing that would be good to explore is whether programs of different sizes fit into different categories and consequently should be used to build different models. For example in the categorical variable model presented above, extremely small and extremely large programs seemed to yield an anomalous results. If these categories do exist, it would be useful to explore where the cutoffs should be. For example, what is the threshold for a large or small program?

The data collection for this thesis effort was problematic in that only a select number of variables were included in the analysis because of the incompleteness of many of the database records. This type of analysis may yield better results if more variables could be included in the analysis. Possibly some of the variables that were not included in the analysis could explain away a great deal of the prediction error.

Still, the results from all the models presented above, both the models of this thesis and the popular models, are less than desirable. Many of these models are based upon similar types of variables, even if they're not used in the same fashion. Because of these results, there could be other variables, other than the variables that are currently being collected that may be useful in predicting software development effort. It would be a worthy effort to explore other possible variables that influence software cost.

Another approach that may yield better results is examining or separating types of applications within command and control. For example within command-and-control there are different types of software development efforts. These types include graphical interfaces, databases, operating systems, diagnostics, message switching, communications, and so forth. In this thesis these categories were not separated. Future efforts could analyze these categories separately.

Finally, linear regression may not be the best method to use with a complex subject like software development. Complex projects like software development are difficult to explain in simple terms. Possibly a more complex, nonlinear model made be more applicable in this area. Still, using a more complex model may not solve the accuracy problem and at the end of the day may not be explainable or understandable to the users of the model.

APPENDICES

Appendix A: Records Selected from SMC and ESC Software Databases

ID	ESLOC	Effort	SLOC	Reused	Months
2517	85,341	196	76,200	13,800	48
2510	43,437	181	43,437		21
348	18,052	418	18,052		12
2502	26,239	744	26,239		59
2505	7,448	211	7,448		59
2506	6,317	179	6,317		59
2508	58,789	1,666	58,789		59
2151	15,025	15,025	15		57
3-2	56,857	406	56,857		24
3-3	16,781	190	16,781		25
3-4	36,688	731	36,688		28
7-1	114,605	1,474	41,946	180,044	28
24-1	21,300	224	21,300		19
24-2	33,400	335	33,400		19
24-3	3,274	81	3,274		19
24-4	53,000	406	53,000		19
24-5	22,000	217	22,000		19
34-1	18,124	230	18,124		25
34-2	27,440	178	27,440		25
34-3	37,183	256	37,183		25
34-4	184,006	1,000	184,006		25
34-5	40,704	217	40,704		25
35-0	26,200	289	26,200		25
36-2	48,156	843	46,382	3,774	21
36-3	20,520	336	20,520		14
36-4	13,366	565	10,456	6,191	23
36-5	90,930	829	86,316	9,817	25
36-6	40,252	1,260	40,252		25
36-7	7,599	374	3,458	8,810	25
36-9	89,506	153	1,245	173,060	19
39-0	46,375	469	36,915	27,029	16
40-3	34,085	97	3,601	48,580	18
40-4	3,330	28	2,300	2,000	18
40-6	2,995	15	1,270	7,500	17
42-1	8,127	120	72	8,689	54
42-2	9,961	64	311	9,772	54

42-3	17,018	99	1,368	13,597	52
44-1	54,192	1,219	54,192		48
44-2	104,090	462	104,090		48
44-3	71,453	181	71,453		48
44-4	75,081	97	75,081		48
51-1	16,375	515	16,375		75
51-2	2,633	148	2,633		75
51-3	22,330	604	22,330		74
51-5	28,192	296	28,192		75
51-6	1,359	32	1,359		74
51-9	1,153	25	1,153		74
51-10	31,574	928	31,574		75

	AC	RV	TC	PE	LE
2517	Nominal	Nominal	Nominal	High	Very High
2510	Nominal	High	Nominal	High	High
348	Nominal	Very High	Extra High	Nominal	Very Low
2502	High	Extra High	Nominal	Nominal	Low
2505	High	Very High	Nominal	Nominal	Low
2506	High	Very High	Nominal	Nominal	Low
2508	High	Very High	Very High	Nominal	Low
2151	Nominal	Nominal	High	Low	Extra High
3-2	Nominal	Nominal	Nominal	High	High
3-3	Nominal	Nominal	Nominal	High	High
3-4	Very High	Nominal	Nominal	High	High
7-1	High	Very Low	Very High	Nominal	Very High
24-1	Very High	Low	Very High	Nominal	Low
24-2	Very High	Low	Nominal	Nominal	Low
24-3	Very High	Low	Very High	Nominal	Low
24-4	Very High	Low	Very High	Nominal	Low
24-5	Very High	Low	Very High	Nominal	Low
34-1	Nominal	Low	Very High	Low	Nominal
34-2	Nominal	Low	Nominal	Low	High
34-3	Nominal	Low	Nominal	Low	High
34-4	Nominal	Low	Very High	Low	Nominal
34-5	Nominal	Low	Very High	Low	Nominal
35-0	Very High	Very Low	Nominal	Very Low	Very Low
36-2	Very High	Nominal	Nominal	Nominal	Extra High
36-3	High	Nominal	Nominal	Nominal	Extra High
36-4	Nominal	Nominal	Nominal	Nominal	Extra High
36-5	Very High	Nominal	Nominal	Nominal	Extra High

36-6	High	Nominal	Nominal	Nominal	Extra High
36-7	High	Nominal	Nominal	Nominal	Extra High
36-9	Nominal	Nominal	Nominal	Nominal	Extra High
39-0	Very High	Low	High	High	Extra High
40-3	Very High	Very Low	Nominal	Low	High
40-4	High	Very Low	Nominal	Very Low	Very Low
40-6	Very High	Very Low	Nominal	Very Low	Low
42-1	Nominal	Nominal	High	Low	High
42-2	High	Nominal	Nominal	Nominal	Extra High
42-3	High	Nominal	High	Low	High
44-1	Very High	Very Low	Nominal	Very High	Extra High
44-2	Very High	Very Low	Nominal	Very High	Extra High
44-3	Very High	Very Low	Nominal	Very High	Extra High
44-4	Extra High	Very Low	Nominal	Very High	Extra High
51-1	High	Very Low	Very High	Low	High
51-2	High	Very Low	Very High	Low	High
51-3	High	Very Low	Very High	Low	High
51-5	High	Very Low	Very High	Low	High
51-6	High	Very Low	Very High	Low	High
51-9	High	Very Low	Very High	Low	High
51-10	High	Very Low	Very High	Low	High

	SL	RSL	MPE	ATS	Language
2517	Nominal	Nominal	Low	Low	0.50
2510	Nominal	Nominal	Nominal	Nominal	-
348	Extra High	Extra High	Nominal	Nominal	-
2502	Very High	Nominal	Low	Nominal	0.05
2505	Very High	Nominal	Low	Nominal	0.05
2506	Very High	Nominal	Low	Nominal	0.05
2508	Very High	Nominal	Low	Nominal	0.05
2151	High	High	High	Nominal	-
3-2	Nominal	Nominal	Nominal	Nominal	-
3-3	Nominal	Nominal	Nominal	Nominal	-
3-4	Nominal	Nominal	Extra High	Nominal	-
7-1	High	Nominal	Extra High	High	0.93
24-1	Nominal	High	High	Nominal	-
24-2	Nominal	High	High	Very High	-
24-3	Nominal	High	High	Very High	-
24-4	Nominal	High	High	Low	-
24-5	Nominal	High	High	Very High	-
34-1	Nominal	High	High	High	-

34-2	Nominal	High	Nominal	High	-
34-3	Nominal	High	Nominal	High	-
34-4	Nominal	High	Nominal	Low	-
34-5	Nominal	High	Nominal	High	-
35-0	Very High	Nominal	Very Low	Nominal	0.07
36-2	High	Nominal	Extra High	High	-
36-3	High	Nominal	Extra High	High	-
36-4	High	Nominal	Extra High	High	-
36-5	High	Nominal	Extra High	High	-
36-6	High	Nominal	Extra High	High	-
36-7	High	Nominal	Extra High	High	-
36-9	High	Nominal	Extra High	High	-
39-0	High	Nominal	Extra High	Very Low	0.55
40-3	Nominal	Nominal	High	Low	1.00
40-4	Nominal	Nominal	Very Low	Nominal	0.40
40-6	Nominal	Nominal	Low	Nominal	1.00
42-1	Nominal	Nominal	High	Low	-
42-2	Very High	Nominal	Extra High	Low	-
42-3	Nominal	Nominal	High	Low	-
44-1	Very High	Nominal	Extra High	High	-
44-2	Very High	Nominal	Extra High	High	0.20
44-3	Very High	Nominal	Extra High	High	0.20
44-4	Very High	Nominal	Extra High	High	0.10
51-1	Very High	Nominal	High	High	0.10
51-2	Very High	Nominal	High	High	0.10
51-3	Very High	Nominal	High	High	0.10
51-5	Very High	Nominal	High	High	0.10
51-6	Very High	Nominal	High	High	0.10
51-9	Very High	Nominal	High	High	0.10
51-10	Very High	Nominal	High	High	0.10

Appendix B. Scatter Plots of Variables vs. Effort

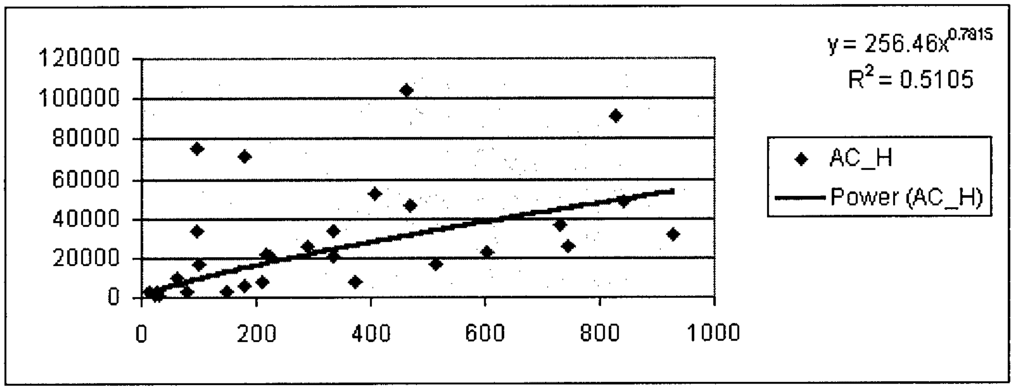


Figure B.1. Scatter Plot of AC_H vs. Effort

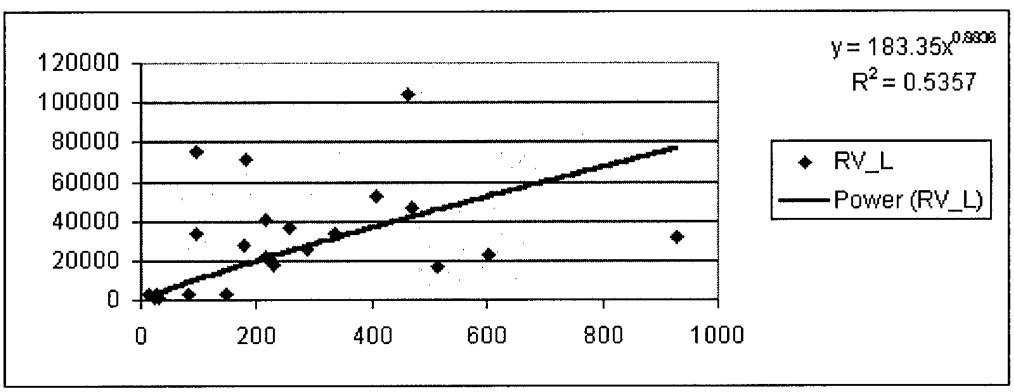


Figure B.2. Scatter Plot of RV_L vs. Effort

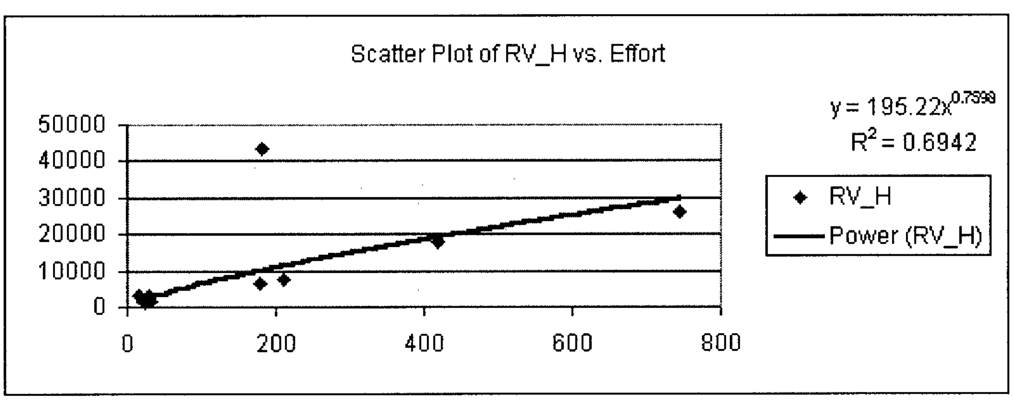


Figure B.3. Scatter Plot of RV_H vs. Effort

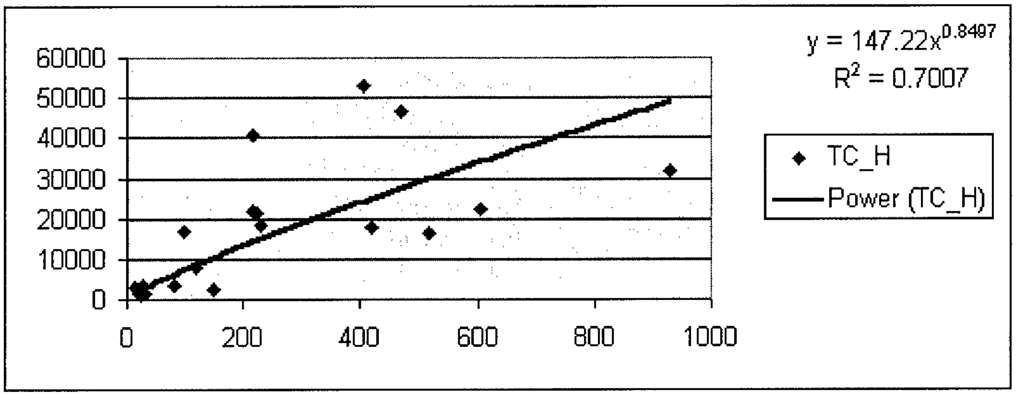


Figure B.4. Scatter Plot of TC_H vs. Effort

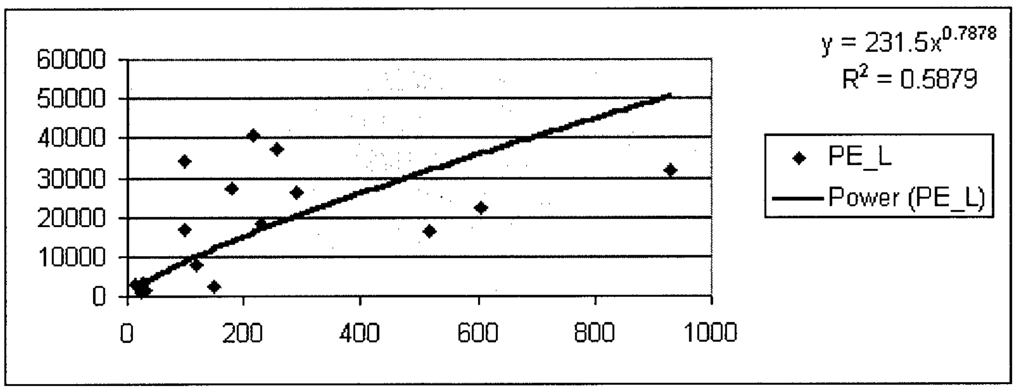


Figure B.5. Scatter Plot of PE_L vs. Effort

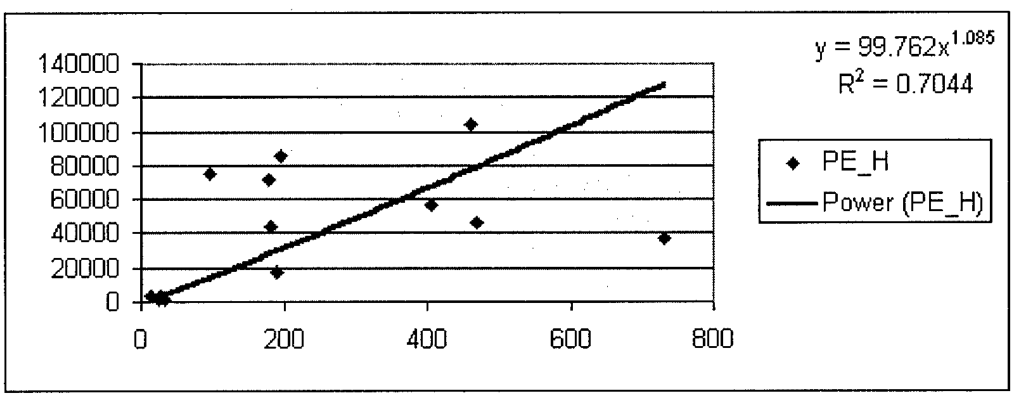


Figure B.6. Scatter Plot of PE_H vs. Effort

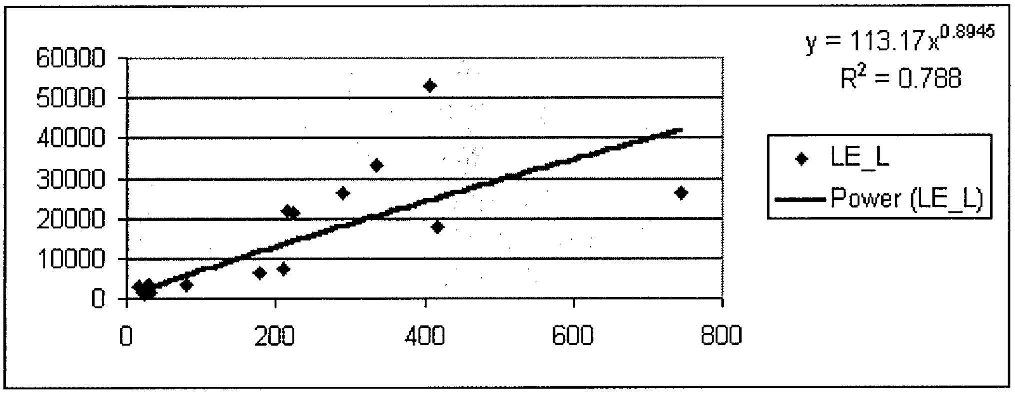


Figure B.7. Scatter Plot of LE_L vs. Effort

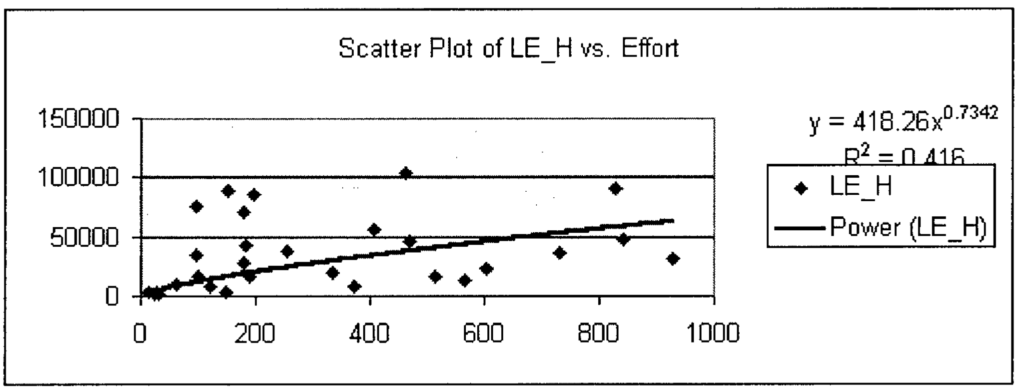


Figure B.8. Scatter Plot of LE_H vs. Effort

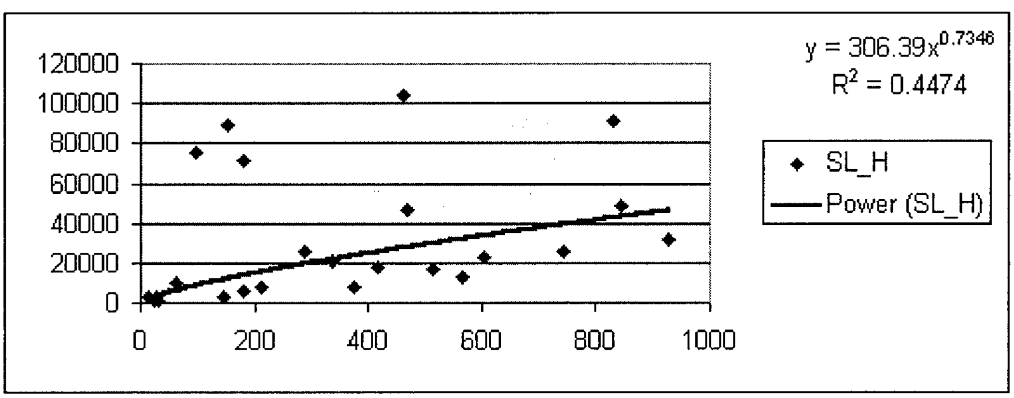


Figure B.9. Scatter Plot of SL_H vs. Effort

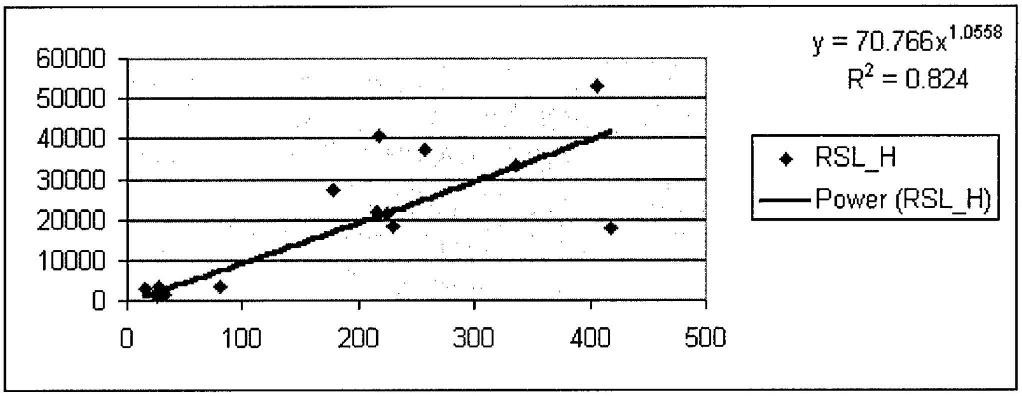


Figure B.10. Scatter Plot of RSL_H vs. Effort

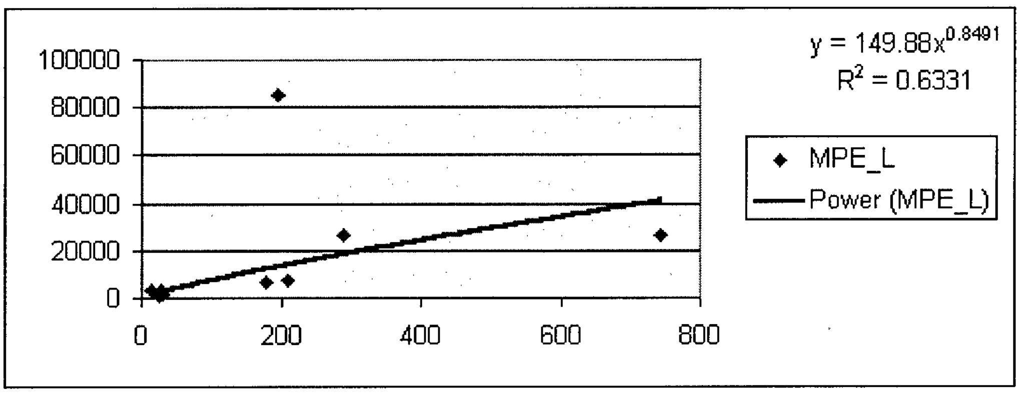


Figure B.11. Scatter Plot of MPE_L vs. Effort

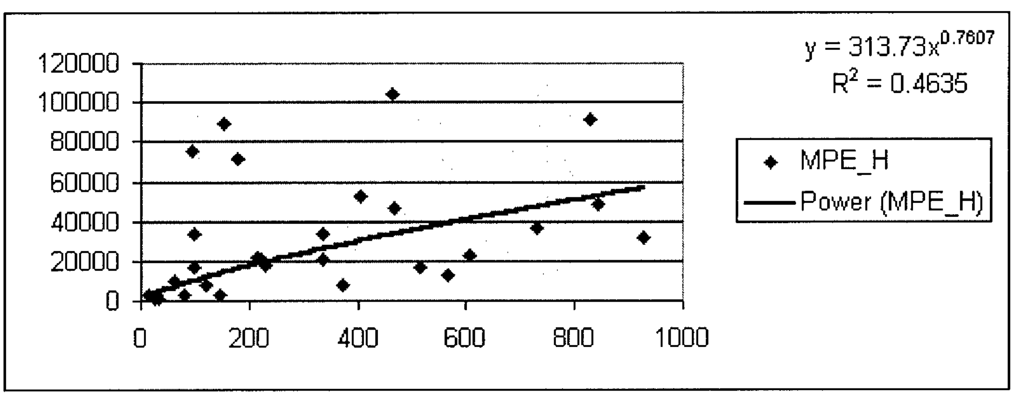


Figure B.12. Scatter Plot of MPE_H vs. Effort

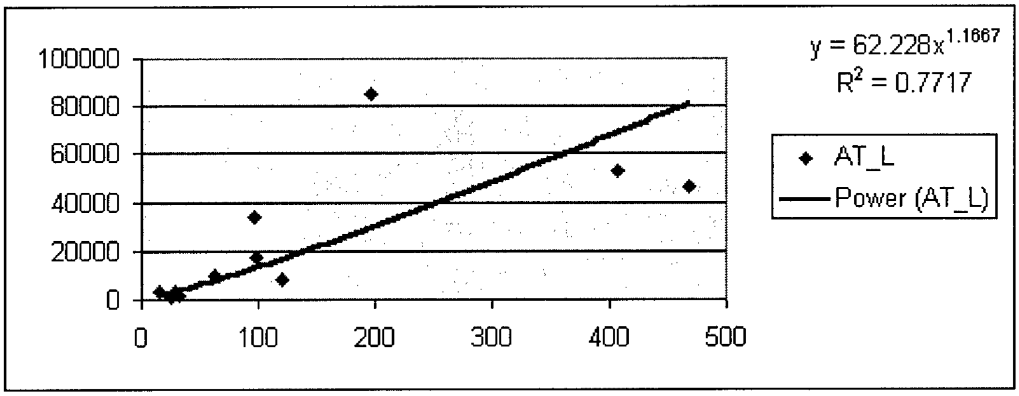


Figure B.13. Scatter Plot of AT_L vs. Effort

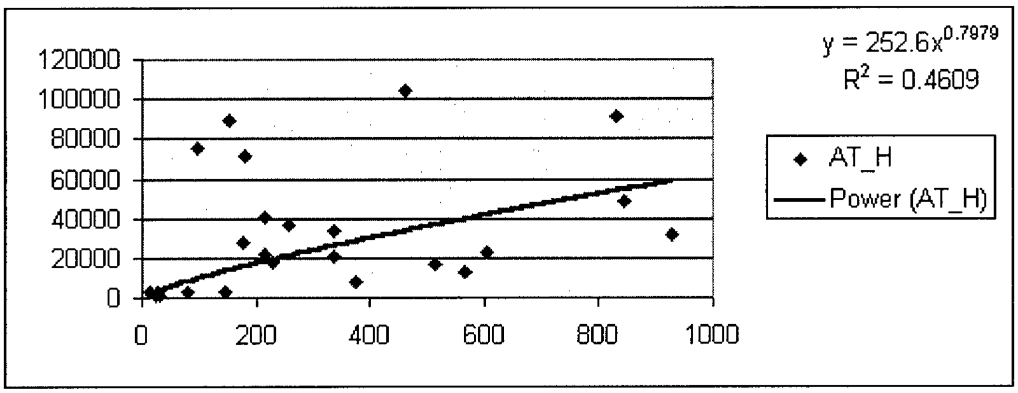


Figure B.14. Scatter Plot of AT_H vs. Effort

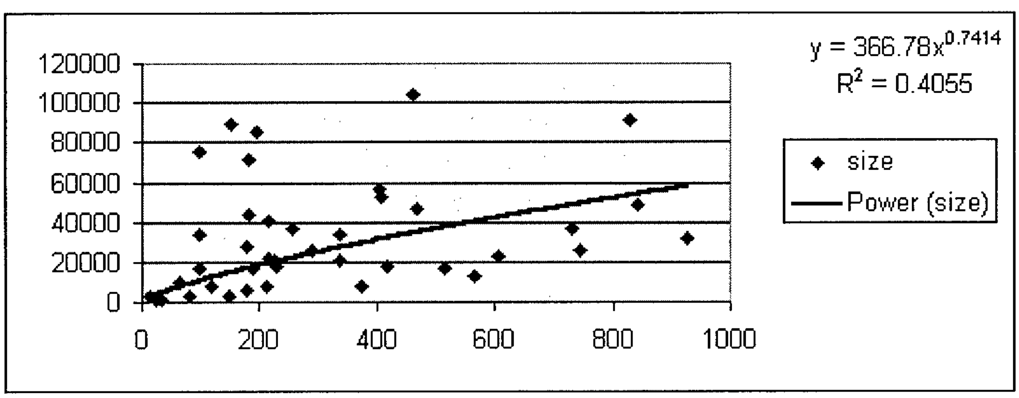


Figure B.15. Scatter Plot of size vs. Effort

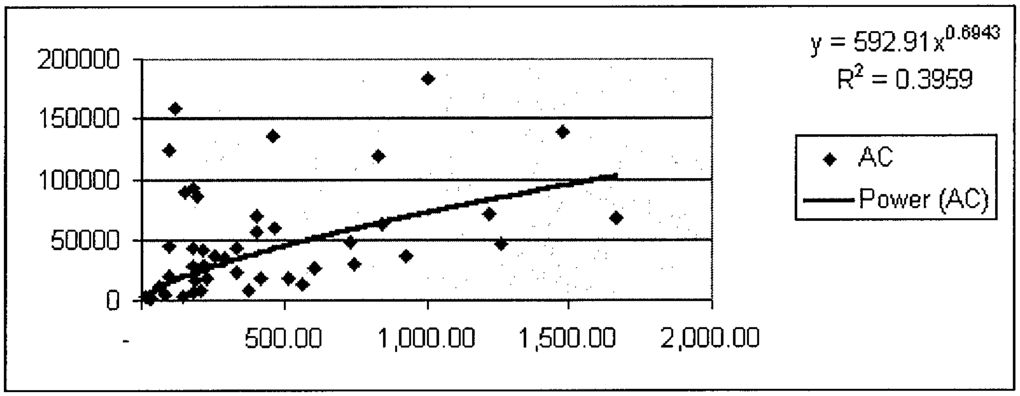


Figure B.16. Scatter Plot of AC vs. Effort

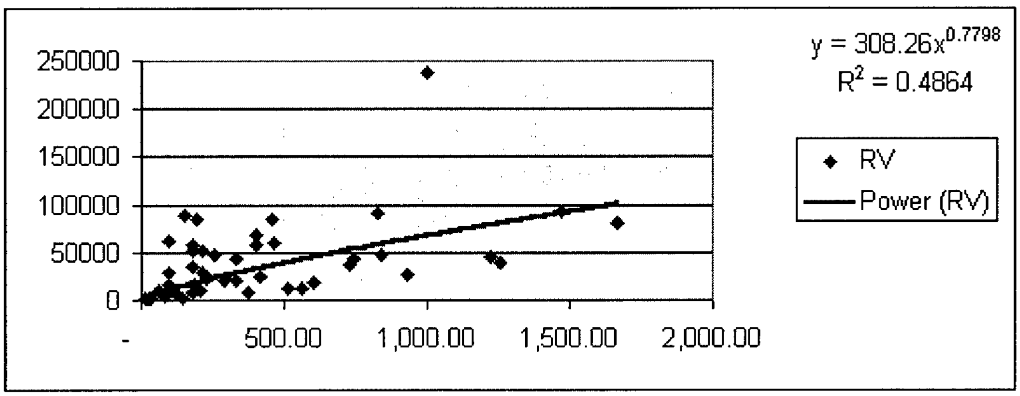


Figure B.17. Scatter Plot of RV vs. Effort

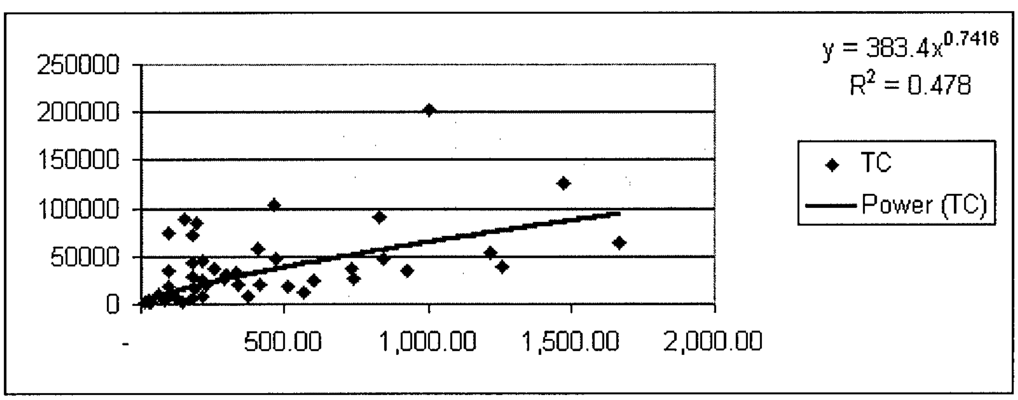


Figure B.18. Scatter Plot of TC vs. Effort

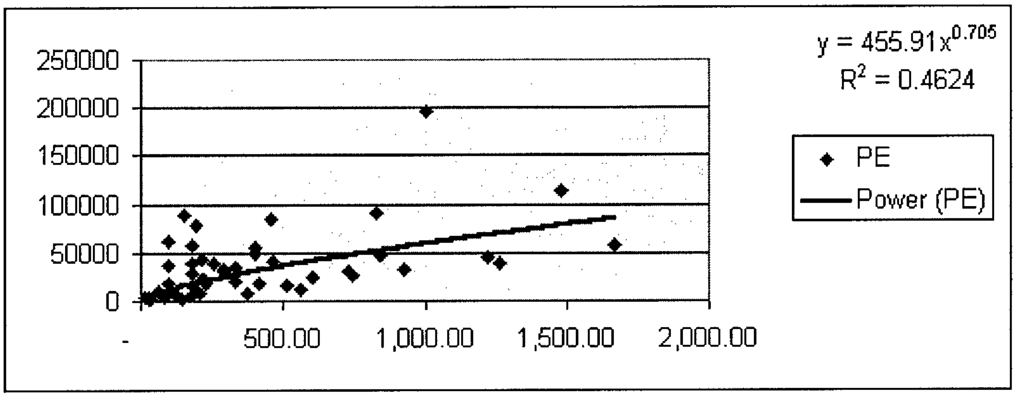


Figure B.19. Scatter Plot of PE vs. Effort

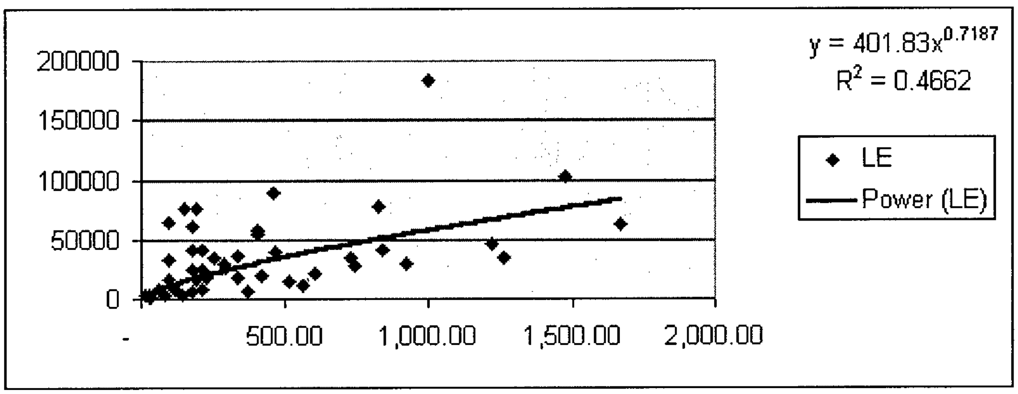


Figure B.20. Scatter Plot of LE vs. Effort

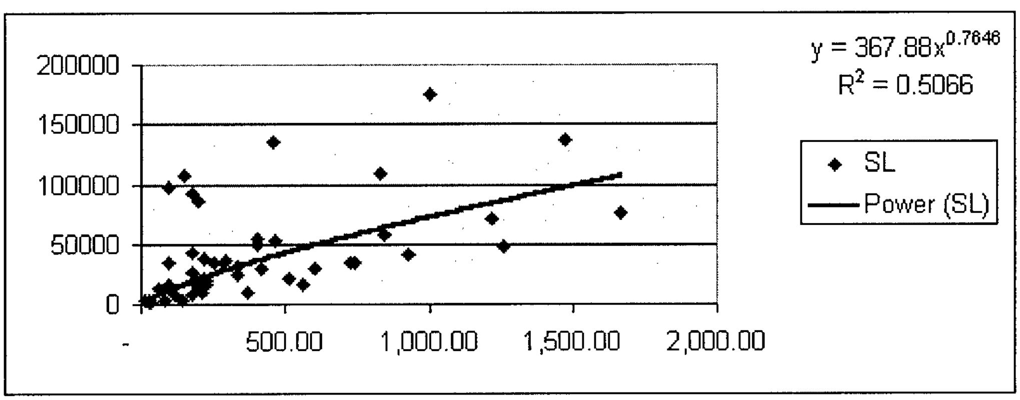


Figure B.21. Scatter Plot of SL vs. Effort

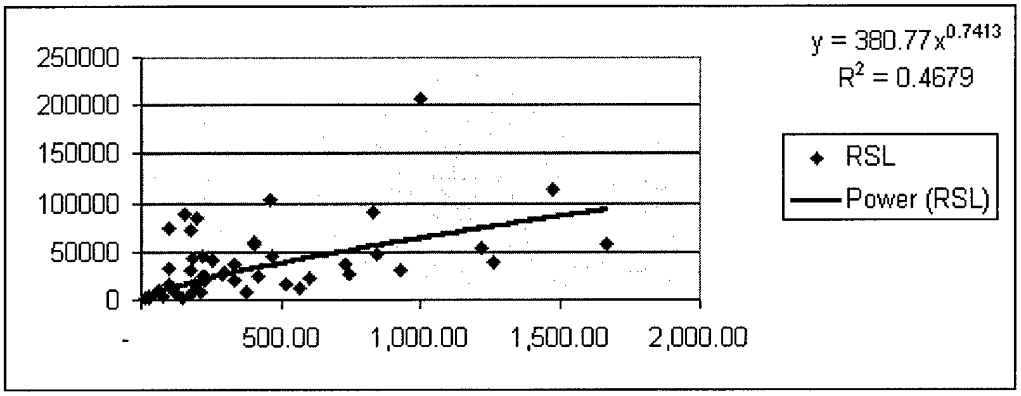


Figure B.22. Scatter Plot of RSL vs. Effort

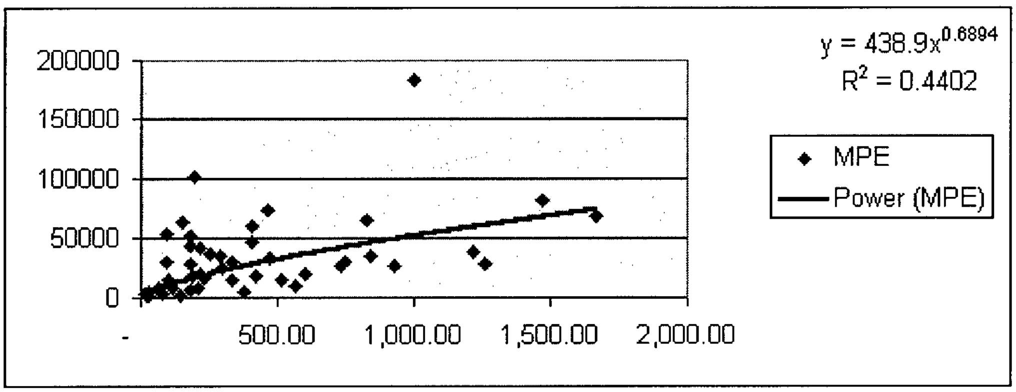


Figure B.23. Scatter Plot of MPE vs. Effort

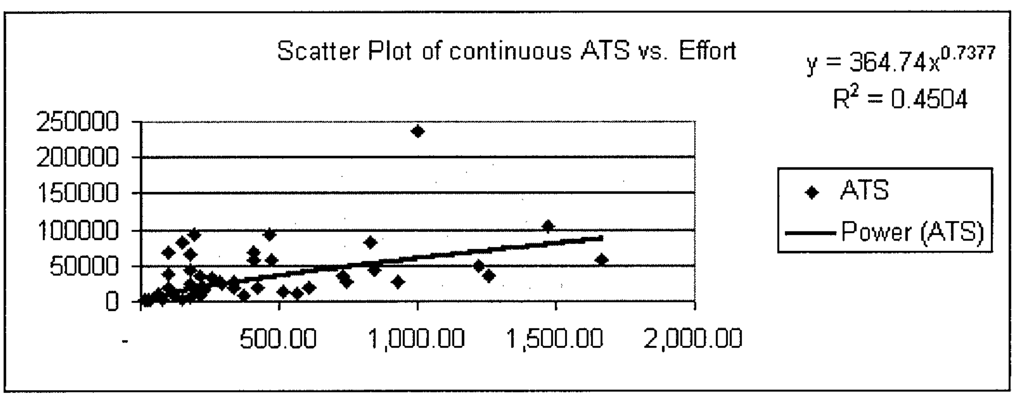


Figure B.24. Scatter Plot of ATS vs. Effort

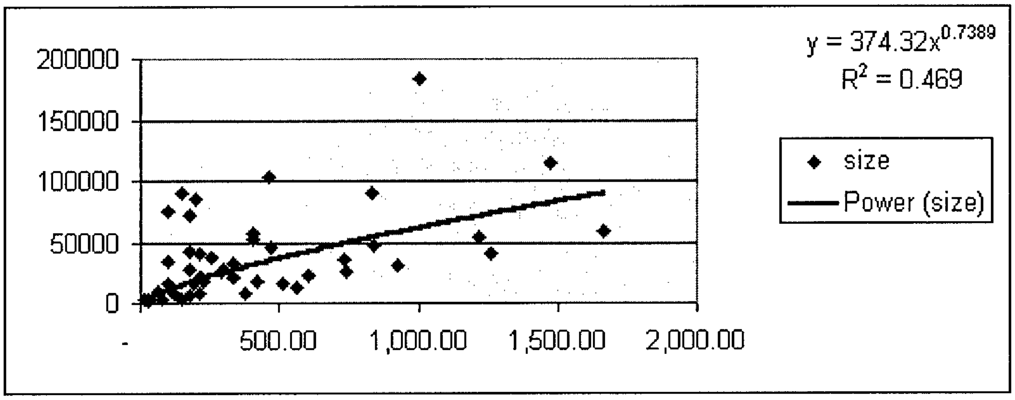


Figure B.25. Scatter Plot of continuous size vs. Effort

Appendix C. Categorical Model Reductions

Table C.1. Categorical Model FULL MODEL

Response:	Effort
Summary of Fit	
RSquare	0.840484
RSquare Adj	0.388522
Root Mean Sq Error	310.251
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	532.149	0.0055	0.942	
SLOC_N	1	1	2323.21	0.0241	0.8791	remove
SLOC_R	1	1	16257.2	0.1689	0.6883	remove
Sched	1	1	67514.8	0.7014	0.4187	
AC_H	1	1	8696.89	0.0904	0.7689	
RV_L	1	1	453.473	0.0047	0.9464	
RV_H	1	1	52755.6	0.5481	0.4733	
TC_H	1	1	18.806	0.0002	0.9891	
PE_L	1	1	17079.5	0.1774	0.681	
PE_H	1	1	3496.39	0.0363	0.852	
LE_L	1	1	25926.1	0.2693	0.6132	
LE_H	1	1	913.204	0.0095	0.924	
SL_H	1	1	23121.6	0.2402	0.6329	
RSL_H	1	1	133.304	0.0014	0.9709	
MPE_L	1	1	4391.67	0.0456	0.8344	
MPE_H	1	1	2386.59	0.0248	0.8775	
AT_L	1	1	11610.6	0.1206	0.7344	
AT_H	1	1	144.201	0.0015	0.9698	
language	1	1	15152.2	0.1574	0.6985	
size^0.75	1	1	28622	0.2974	0.5955	
achp	1	1	17219.6	0.1789	0.6798	
rvlp	1	1	8347.54	0.0867	0.7734	
tchp	1	1	10201.8	0.106	0.7504	
pelp	1	1	32531.7	0.338	0.5718	
lelp	1	1	28659.7	0.2977	0.5953	
atlp	1	1	15451.3	0.1605	0.6957	
athp	1	1	11529.9	0.1198	0.7353	
AC_H*Norm_siz	1	1	21383.7	0.2222	0.6459	
TC_H*Norm_siz	1	1	10570.9	0.1098	0.7461	
PE_L*Norm_siz	1	1	34543.2	0.3589	0.5603	
LE_L*Norm_siz	1	1	27720.5	0.288	0.6013	
AT_L*Norm_siz	1	1	14723.8	0.153	0.7026	
AT_H*Norm_siz	1	1	16468.9	0.1711	0.6864	

RV_L*Norm_siz	1	1	8827.29	0.0917	0.7672

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	34	6086006	179000	1.8596
Error	12	1155068	96256	Prob>F
C Total	46	7241074		0.1256

Table C.2. Categorical Model REDUCTION 1

Response:	Effort
Summary of Fit	
RSquare	0.835487
RSquare Adj	0.459458
Root Mean Sq Error	291.7005
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	96.593	0.0011	0.9736	
Sched	1	1	40552	0.4766	0.5013	
AC_H	1	1	25569	0.3005	0.5922	
RV_L	1	1	3079.71	0.0362	0.8518	
RV_H	1	1	17908.8	0.2105	0.6534	
TC_H	1	1	6.35	0.0001	0.9932	
PE_L	1	1	787.942	0.0093	0.9247	
PE_H	1	1	143.901	0.0017	0.9678	remove
LE_L	1	1	9856.28	0.1158	0.7386	
LE_H	1	1	10127.8	0.119	0.7352	
SL_H	1	1	56303.4	0.6617	0.4296	
RSL_H	1	1	41122.6	0.4833	0.4983	
MPE_L	1	1	18522.6	0.2177	0.648	
MPE_H	1	1	1346.19	0.0158	0.9017	
AT_L	1	1	3253.45	0.0382	0.8478	
AT_H	1	1	333.888	0.0039	0.9509	
language	1	1	4156.01	0.0488	0.8283	
size^0.75	1	1	834.584	0.0098	0.9225	
achp	1	1	9427.26	0.1108	0.7442	
rvlp	1	1	16517	0.1941	0.6662	
tchp	1	1	14413.9	0.1694	0.6869	
pelp	1	1	1754.76	0.0206	0.8879	
lelp	1	1	9202.54	0.1082	0.7471	
atlp	1	1	2318.75	0.0273	0.8712	
athp	1	1	614.587	0.0072	0.9335	
AC_H*Norm_siz	1	1	5880.31	0.0691	0.7965	
TC_H*Norm_siz	1	1	35066.9	0.4121	0.5313	
PE_L*Norm_siz	1	1	1820.43	0.0214	0.8858	
LE_L*Norm_siz	1	1	7885.2	0.0927	0.7653	
AT_L*Norm_siz	1	1	3096.45	0.0364	0.8514	

AT_H*Norm_siz	1	1	1223.55	0.0144	0.9063
RV_L*Norm_siz	1	1	14676.9	0.1725	0.6842

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	32	6049826	189057	2.2219
Error	14	1191249	85089	Prob>F
C Total	46	7241074		0.057

SSEfull	1155068	Test Stat	0.18794
Betafull	34		
dffull	12	F2,12	3.88529
SSEred	1191249		
Betared	32	Conclude:	Reduced at least as good

Table C.3. Categorical Model REDUCTION 2

Response:	Effort
Summary of Fit	
RSquare	0.835467
RSquare Adj	0.495433
Root Mean Sq Error	281.8265
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F
Norm_siz	1	1	169.973	0.0021	0.9637
Sched	1	1	41586.6	0.5236	0.4804
AC_H	1	1	32202	0.4054	0.5339
RV_L	1	1	2966.2	0.0373	0.8494
RV_H	1	1	18044.8	0.2272	0.6405
TC_H	1	1	10.589	0.0001	0.9909
PE_L	1	1	874.426	0.011	0.9178
LE_L	1	1	10205.3	0.1285	0.725
LE_H	1	1	10175.9	0.1281	0.7254
SL_H	1	1	78862.3	0.9929	0.3348
RSL_H	1	1	43831.9	0.5519	0.469
MPE_L	1	1	18732.3	0.2358	0.6342
MPE_H	1	1	1361.61	0.0171	0.8976
AT_L	1	1	3111.02	0.0392	0.8458
AT_H	1	1	528.126	0.0066	0.9361
language	1	1	4013.95	0.0505	0.8252
size^0.75	1	1	816.795	0.0103	0.9206
achp	1	1	12499.8	0.1574	0.6972
rvlp	1	1	17402.4	0.2191	0.6465
tchp	1	1	14748.4	0.1857	0.6727
pelp	1	1	2105.02	0.0265	0.8729

lelp	1	1	11992.2	0.151	0.7031
atlp	1	1	2175.1	0.0274	0.8708
athp	1	1	486.831	0.0061	0.9386remove
AC_H*Norm_siz	1	1	7968.74	0.1003	0.7558
TC_H*Norm_siz	1	1	38460.7	0.4842	0.4972
PE_L*Norm_siz	1	1	1874.23	0.0236	0.88
LE_L*Norm_siz	1	1	9759.54	0.1229	0.7308
AT_L*Norm_siz	1	1	2953.34	0.0372	0.8497
AT_H*Norm_siz	1	1	1081.47	0.0136	0.9087
RV_L*Norm_siz	1	1	15219.8	0.1916	0.6678

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	31	6049682	195151	2.457
Error	15	1191393	79426	Prob>F
C Total	46	7241074		0.034

SSEfull	1191249	Test Stat	0.00169
Betafull	32		
dffull	14	F1,14	4.60011
SSEred	1191393		
Betared	31	Conclude:	Reduced at least as good

Table C.4. Categorical Model REDUCTION 3

Response:	Effort
Summary of Fit	
RSquare	0.8354
RSquare Adj	0.526775
Root Mean Sq Error	272.9331
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F
Norm_siz	1	1	1161.92	0.0156	0.9022
Sched	1	1	57577.9	0.7729	0.3923
AC_H	1	1	32711.1	0.4391	0.517
RV_L	1	1	2557.17	0.0343	0.8553
RV_H	1	1	20651.5	0.2772	0.6057
TC_H	1	1	505.537	0.0068	0.9354
PE_L	1	1	6698.17	0.0899	0.7681
LE_L	1	1	42643.7	0.5725	0.4603
LE_H	1	1	9757.14	0.131	0.7222
SL_H	1	1	79094.8	1.0618	0.3181
RSL_H	1	1	47267.8	0.6345	0.4374
MPE_L	1	1	18309.5	0.2458	0.6268
MPE_H	1	1	1109.02	0.0149	0.9044remove

AT_L	1	1	4459.8	0.0599	0.8098
AT_H	1	1	27331.8	0.3669	0.5532
language	1	1	5947.84	0.0798	0.7811
size^0.75	1	1	3186.46	0.0428	0.8388
achp	1	1	12864.9	0.1727	0.6832
rvlp	1	1	19391.3	0.2603	0.6169
tchp	1	1	19021.9	0.2554	0.6202
pelp	1	1	7494.1	0.1006	0.7552
lelp	1	1	40718.7	0.5466	0.4704
atlp	1	1	2245.88	0.0301	0.8643
AC_H*Norm_siz	1	1	8262.88	0.1109	0.7434
TC_H*Norm_siz	1	1	52017	0.6983	0.4157
PE_L*Norm_siz	1	1	4602.34	0.0618	0.8069
LE_L*Norm_siz	1	1	26326.9	0.3534	0.5605
AT_L*Norm_siz	1	1	3493.3	0.0469	0.8313
AT_H*Norm_siz	1	1	29391.8	0.3946	0.5388
RV_L*Norm_siz	1	1	16713.8	0.2244	0.6421

Whole-Model Test
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	30	6049195	201640	2.7068
Error	16	1191880	74492	Prob>F
C Total	46	7241074		0.0195

SSEfull	1191393	Test Stat	0.00613
Betafull	31		
dffull	15	F1,15	4.54307
SSEred	1191880		
Betared	30	Conclude:	Reduced at least as good

Table C.5. Categorical Model REDUCTION 4

Response: Effort

Summary of Fit

RSquare	0.835247
RSquare Adj	0.554198
Root Mean Sq Error	264.9072
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F
Norm_siz	1	1	1826.34	0.026	0.8737
Sched	1	1	60791.4	0.8663	0.365
AC_H	1	1	87393.7	1.2454	0.28
RV_L	1	1	3077.5	0.0439	0.8366
RV_H	1	1	20046.7	0.2857	0.5999
TC_H	1	1	1086.16	0.0155	0.9025

PE_L	1	1	5862.8	0.0835	0.776	
LE_L	1	1	44147.2	0.6291	0.4386	
LE_H	1	1	8791.46	0.1253	0.7277	
SL_H	1	1	78053.3	1.1123	0.3064	
RSL_H	1	1	50732.2	0.7229	0.407	
MPE_L	1	1	20517.4	0.2924	0.5957	
AT_L	1	1	5247.38	0.0748	0.7878	
AT_H	1	1	44181.2	0.6296	0.4384	
language	1	1	5632.28	0.0803	0.7804	
size^0.75	1	1	2394.01	0.0341	0.8556	remove
achp	1	1	54516.3	0.7769	0.3904	
rvlp	1	1	19992.3	0.2849	0.6004	
tchp	1	1	17969.6	0.2561	0.6193	
pelp	1	1	6446.76	0.0919	0.7655	
lelp	1	1	39846.1	0.5678	0.4614	
atlp	1	1	3969.84	0.0566	0.8148	
AC_H*Norm_siz	1	1	38388.2	0.547	0.4696	
TC_H*Norm_siz	1	1	50917.4	0.7256	0.4062	
PE_L*Norm_siz	1	1	3519.24	0.0501	0.8255	
LE_L*Norm_siz	1	1	25318	0.3608	0.556	
AT_L*Norm_siz	1	1	5853.45	0.0834	0.7762	
AT_H*Norm_siz	1	1	38721.2	0.5518	0.4677	
RV_L*Norm_siz	1	1	17185.6	0.2449	0.627	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	29	6048086	208555	2.9719
Error	17	1192989	70176	Prob>F
C Total	46	7241074		0.0108

SSEfull	1191880	Test Stat	0.01489
Betafull	30		
dffull	16	F1,16	4.494
SSEred	1192989		
Betared	29	Conclude:	Reduced at least as good

Table C.6. Categorical Model REDUCTION 5

Response:	Effort
Summary of Fit	
RSquare	0.834916
RSquare Adj	0.57812
Root Mean Sq Error	257.7017
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	1563.49	0.0235	0.8798	
Sched	1	1	64165.5	0.9662	0.3387	
AC_H	1	1	347619	5.2344	0.0345	
RV_L	1	1	3403.11	0.0512	0.8235	
RV_H	1	1	17729.8	0.267	0.6117	
TC_H	1	1	4146.03	0.0624	0.8055	
PE_L	1	1	6151.53	0.0926	0.7644	
LE_L	1	1	41834.6	0.6299	0.4377	
LE_H	1	1	10500.9	0.1581	0.6956	
SL_H	1	1	75867.5	1.1424	0.2993	
RSL_H	1	1	83736.1	1.2609	0.2762	
MPE_L	1	1	23801.8	0.3584	0.5569	
AT_L	1	1	11913.4	0.1794	0.6769	
AT_H	1	1	54176.9	0.8158	0.3783	
language	1	1	4280.43	0.0645	0.8025	remove
achp	1	1	344825	5.1924	0.0351	
rvlp	1	1	19626.3	0.2955	0.5934	
tchp	1	1	15606.4	0.235	0.6337	
pelp	1	1	5718.86	0.0861	0.7725	
lelp	1	1	37468	0.5642	0.4623	
atlp	1	1	11438.1	0.1722	0.683	
AC_H*Norm_siz	1	1	244670	3.6842	0.0709	
TC_H*Norm_siz	1	1	49760	0.7493	0.3981	
PE_L*Norm_siz	1	1	2768.85	0.0417	0.8405	
LE_L*Norm_siz	1	1	22924	0.3452	0.5641	
AT_L*Norm_siz	1	1	15976.3	0.2406	0.6297	
AT_H*Norm_siz	1	1	66245	0.9975	0.3312	
RV_L*Norm_siz	1	1	16780.5	0.2527	0.6213	

Whole-Model Test

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	28	6045692	215918	3.2513
Error	18	1195383	66410	Prob>F
C Total	46	7241074		0.0057

SSEfull	1192989	Test Stat	0.03411
Betafull	29		
dffull	17	F1,17	4.45132
SSEred	1195383		
Betared	28	Conclude:	Reduced at least as good

Table C.7. Categorical Model REDUCTION 6

Response:	Effort
Summary of Fit	
RSquare	0.834325
RSquare Adj	0.598893
Root Mean Sq Error	251.2771
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	1093.1	0.0173	0.8967	
Sched	1	1	61094.2	0.9676	0.3376	
AC_H	1	1	370284	5.8645	0.0256	
RV_L	1	1	8657.47	0.1371	0.7153	
RV_H	1	1	14545.1	0.2304	0.6367	
TC_H	1	1	7352.98	0.1165	0.7367	
PE_L	1	1	5840.79	0.0925	0.7643	
LE_L	1	1	41289.3	0.6539	0.4287	
LE_H	1	1	11724	0.1857	0.6714	
SL_H	1	1	71779.2	1.1368	0.2997	
RSL_H	1	1	79700.2	1.2623	0.2752	
MPE_L	1	1	25430	0.4028	0.5332	
AT_L	1	1	13259	0.21	0.652	
AT_H	1	1	58374.6	0.9245	0.3484	
achp	1	1	353124	5.5927	0.0288	
rvlp	1	1	28020.6	0.4438	0.5133	
tchp	1	1	13709.7	0.2171	0.6465	
pelp	1	1	4970.82	0.0787	0.7821	remove
lelp	1	1	33464.2	0.53	0.4755	
atlp	1	1	15833.9	0.2508	0.6223	
AC_H*Norm_siz	1	1	248626	3.9377	0.0618	
TC_H*Norm_siz	1	1	48145.1	0.7625	0.3934	
PE_L*Norm_siz	1	1	2243.58	0.0355	0.8525	
LE_L*Norm_siz	1	1	19088.5	0.3023	0.5888	
AT_L*Norm_siz	1	1	20824.6	0.3298	0.5725	
AT_H*Norm_siz	1	1	66127.1	1.0473	0.319	
RV_L*Norm_siz	1	1	24148.4	0.3825	0.5436	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	27	6041411	223756	3.5438
Error	19	1199663	63140	Prob>F
C Total	46	7241074		0.003

SSEfull	1195383	Test Stat	0.06445

Betafull	28		
dffull	18	F1,18	4.41386
SSEred	1199663		
Betared	27	Conclude:	Reduced at least as good

Table C.8. Categorical Model REDUCTION 7

Response:	Effort
Summary of Fit	
RSquare	0.833639
RSquare Adj	0.617369
Root Mean Sq Error	245.4215
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	16810.6	0.2791	0.6031	
Sched	1	1	56144.4	0.9321	0.3458	
AC_H	1	1	366891	6.0913	0.0227	
RV_L	1	1	16027.2	0.2661	0.6116	
RV_H	1	1	10451.9	0.1735	0.6814	
TC_H	1	1	6554.67	0.1088	0.7449	
PE_L	1	1	1369.39	0.0227	0.8817	
LE_L	1	1	37247.7	0.6184	0.4409	
LE_H	1	1	8848.57	0.1469	0.7056	
SL_H	1	1	75095.7	1.2468	0.2774	
RSL_H	1	1	74908.9	1.2437	0.278	
MPE_L	1	1	20715.8	0.3439	0.5641	
AT_L	1	1	8900.63	0.1478	0.7047	
AT_H	1	1	53793.8	0.8931	0.3559	
achp	1	1	349271	5.7988	0.0258	
rvlp	1	1	37780.7	0.6273	0.4377	
tchp	1	1	16739	0.2779	0.6039	
lelp	1	1	36806.1	0.6111	0.4435	
atlp	1	1	10971.8	0.1822	0.6741	
AC_H*Norm_siz	1	1	246650	4.095	0.0566	
TC_H*Norm_siz	1	1	53745.4	0.8923	0.3561	
PE_L*Norm_siz	1	1	6551.98	0.1088	0.745	remove
LE_L*Norm_siz	1	1	23945.8	0.3976	0.5355	
AT_L*Norm_siz	1	1	16837.7	0.2795	0.6028	
AT_H*Norm_siz	1	1	61259.5	1.0171	0.3253	
RV_L*Norm_siz	1	1	32936.9	0.5468	0.4682	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	26	6036441	232171	3.8546
Error	20	1204634	60232	Prob>F
C Total	46	7241074		0.0015

SSEfull	1199663	Test Stat	0.07873
Betafull	27		
dffull	19	F1,19	4.38075
SSEred	1204634		
Betared	26	Conclude:	Reduced at least as good

Table C.9. Categorical Model REDUCTION 8

Response:	Effort
Summary of Fit	
RSquare	0.832734
RSquare Adj	0.633608
Root Mean Sq Error	240.1573
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	18592.8	0.3224	0.5762	
Sched	1	1	54383.2	0.9429	0.3426	
AC_H	1	1	374719	6.497	0.0187	
RV_L	1	1	33729.1	0.5848	0.4529	
RV_H	1	1	12054.9	0.209	0.6522	
TC_H	1	1	4845.41	0.084	0.7748	
PE_L	1	1	5498.28	0.0953	0.7605	remove
LE_L	1	1	30819.4	0.5344	0.4729	
LE_H	1	1	15000.9	0.2601	0.6154	
SL_H	1	1	68773.4	1.1924	0.2872	
RSL_H	1	1	68545.7	1.1885	0.288	
MPE_L	1	1	40432.9	0.701	0.4119	
AT_L	1	1	14414.5	0.2499	0.6223	
AT_H	1	1	48283	0.8371	0.3706	
achp	1	1	342747	5.9427	0.0238	
rvlp	1	1	55493.3	0.9622	0.3378	
tchp	1	1	14380.2	0.2493	0.6227	
lelp	1	1	31299.9	0.5427	0.4695	
atlp	1	1	21722.9	0.3766	0.546	
AC_H*Norm_siz	1	1	240135	4.1636	0.0541	
TC_H*Norm_siz	1	1	48688.7	0.8442	0.3686	
LE_L*Norm_siz	1	1	20370.5	0.3532	0.5587	
AT_L*Norm_siz	1	1	30502.9	0.5289	0.4751	
AT_H*Norm_siz	1	1	55234.8	0.9577	0.3389	
RV_L*Norm_siz	1	1	48259.1	0.8367	0.3707	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	25	6029889	241196	4.1819

Error	21 1211186	57676	Prob>F
C Total	46 7241074		0.0007

SSEfull	1204634	Test Stat	0.10878
Betafull	26		
dffull	20	F1,20	4.35125
SSEred	1211186		
Betared	25	Conclude:	Reduced at least as good

Table C.10. Categorical Model REDUCTION 9

Response:	Effort
Summary of Fit	
RSquare	0.831975
RSquare Adj	0.648674
Root Mean Sq Error	235.1677
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	13916.3	0.2516	0.6209	
Sched	1	1	58914	1.0653	0.3132	
AC_H	1	1	369479	6.6809	0.0169	
RV_L	1	1	28230.8	0.5105	0.4824	
RV_H	1	1	15827.2	0.2862	0.598	
TC_H	1	1	6281.84	0.1136	0.7393	
LE_L	1	1	52168.8	0.9433	0.342	
LE_H	1	1	13352.7	0.2414	0.628	remove
SL_H	1	1	80147.1	1.4492	0.2414	
RSL_H	1	1	67018.1	1.2118	0.2829	
MPE_L	1	1	35358.6	0.6394	0.4325	
AT_L	1	1	22307.6	0.4034	0.5319	
AT_H	1	1	61118.9	1.1051	0.3045	
achp	1	1	339930	6.1466	0.0213	
rvlp	1	1	51928.9	0.939	0.3431	
tchp	1	1	13984.4	0.2529	0.6201	
lelp	1	1	32391.8	0.5857	0.4522	
atlp	1	1	31466.4	0.569	0.4587	
AC_H*Norm_siz	1	1	236166	4.2703	0.0508	
TC_H*Norm_siz	1	1	47995.2	0.8678	0.3617	
LE_L*Norm_siz	1	1	20202.8	0.3653	0.5518	
AT_L*Norm_siz	1	1	41920.1	0.758	0.3934	
AT_H*Norm_siz	1	1	80052.1	1.4475	0.2417	
RV_L*Norm_siz	1	1	45085.3	0.8152	0.3764	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio

Model	24 6024390	251016	4.5389
Error	22 1216684	55304	Prob>F
C Total	46 7241074		0.0003

SSEfull	1211186	Test Stat	0.09533
Betafull	25		
dffull	21	F1,21	4.32479
SSEred	1216684		
Betared	24	Conclude:	Reduced at least as good

Table C.11. Categorical Model REDUCTION 10

Response:	Effort
Summary of Fit	
RSquare	0.830131
RSquare Adj	0.660261
Root Mean Sq Error	231.2571
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	8060.19	0.1507	0.7014	
Sched	1	1	96168.9	1.7982	0.193	
AC_H	1	1	394667	7.3797	0.0123	
RV_L	1	1	29365.2	0.5491	0.4662	
RV_H	1	1	22492.5	0.4206	0.5231	
TC_H	1	1	8780.54	0.1642	0.6891	
LE_L	1	1	45630.2	0.8532	0.3652	
SL_H	1	1	84789.4	1.5854	0.2206	
RSL_H	1	1	54809.5	1.0249	0.3219	
MPE_L	1	1	22374.3	0.4184	0.5242	
AT_L	1	1	29415	0.55	0.4658	
AT_H	1	1	67303.1	1.2585	0.2735	
achp	1	1	341333	6.3825	0.0189	
rvlp	1	1	51639.3	0.9656	0.336	
tchp	1	1	17664.3	0.3303	0.5711	remove
lelp	1	1	30816.7	0.5762	0.4555	
atlp	1	1	47747.8	0.8928	0.3545	
AC_H*Norm_siz	1	1	234830	4.391	0.0473	
TC_H*Norm_siz	1	1	57107.1	1.0678	0.3122	
LE_L*Norm_siz	1	1	18072.4	0.3379	0.5667	
AT_L*Norm_siz	1	1	60580.1	1.1328	0.2982	
AT_H*Norm_siz	1	1	91827.3	1.717	0.203	
RV_L*Norm_siz	1	1	44359.1	0.8295	0.3719	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	23	6011038	261349	4.8869
Error	23	1230037	53480	Prob>F
C Total	46	7241074		0.0002

SSEfull	1216684	Test Stat	0.24144
Betafull	24		
dffull	22	F1,22	4.30094
SSEred	1230037		
Betared	23	Conclude:	Reduced at least as good

Table C.12. Categorical Model REDUCTION 11

Response:	Effort
Summary of Fit	
RSquare	0.827691
RSquare Adj	0.669741
Root Mean Sq Error	228.0078
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	13359.4	0.257	0.6168	
Sched	1	1	115613	2.2239	0.1489	
AC_H	1	1	395432	7.6063	0.0109	
RV_L	1	1	93814.6	1.8046	0.1917	
RV_H	1	1	17588.3	0.3383	0.5662	
TC_H	1	1	193315	3.7185	0.0657	
LE_L	1	1	35807.2	0.6888	0.4148	
SL_H	1	1	83221.3	1.6008	0.2179	
RSL_H	1	1	53305.5	1.0254	0.3214	
MPE_L	1	1	14368	0.2764	0.6039	remove
AT_L	1	1	18909.9	0.3637	0.5521	
AT_H	1	1	109959	2.1151	0.1588	
achp	1	1	328399	6.3169	0.0191	
rvlp	1	1	130583	2.5118	0.1261	
lelp	1	1	16774.9	0.3227	0.5753	
atlp	1	1	42471.4	0.817	0.3751	
AC_H*Norm_siz	1	1	222389	4.2777	0.0496	
TC_H*Norm_siz	1	1	660332	12.7017	0.0016	
LE_L*Norm_siz	1	1	6860.56	0.132	0.7196	
AT_L*Norm_siz	1	1	56109.3	1.0793	0.3092	
AT_H*Norm_siz	1	1	127028	2.4434	0.1311	
RV_L*Norm_siz	1	1	114779	2.2078	0.1503	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	22	5993373	272426	5.2402
Error	24	1247701	51988	Prob>F
C Total	46	7241074		<.0001

SSEfull	1230037	Test Stat	0.3303
Betafull	23		
dffull	23	F1,23	4.27934
SSEred	1247701		
Betared	22	Conclude:	Reduced at least as good

Table C.13. Categorical Model REDUCTION 12

Response: Effort

Summary of Fit

RSquare	0.825707
RSquare Adj	0.679301
Root Mean Sq Error	224.6837
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	12058.8	0.2389	0.6293	
Sched	1	1	170636	3.3801	0.0779	
AC_H	1	1	401042	7.9441	0.0093	
RV_L	1	1	95238.1	1.8865	0.1818	
RV_H	1	1	15718.5	0.3114	0.5818	remove
TC_H	1	1	302574	5.9936	0.0217	
LE_L	1	1	77503.9	1.5353	0.2268	
SL_H	1	1	79539.9	1.5756	0.221	
RSL_H	1	1	40266.3	0.7976	0.3803	
AT_L	1	1	20085.4	0.3979	0.5339	
AT_H	1	1	106291	2.1055	0.1592	
achp	1	1	314425	6.2284	0.0195	
rvlp	1	1	119067	2.3586	0.1372	
lelp	1	1	19617.4	0.3886	0.5387	
atlp	1	1	40318.5	0.7987	0.38	
AC_H*Norm_siz	1	1	208807	4.1362	0.0527	
TC_H*Norm_siz	1	1	859671	17.029	0.0004	
LE_L*Norm_siz	1	1	8162.69	0.1617	0.691	
AT_L*Norm_siz	1	1	52141	1.0328	0.3192	
AT_H*Norm_siz	1	1	123816	2.4526	0.1299	
RV_L*Norm_siz	1	1	103001	2.0403	0.1656	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	21	5979005	284715	5.6398
Error	25	1262069	50483	Prob>F
C Total	46	7241074		<.0001

SSEfull	1247701	Test Stat	0.27637
Betafull	22		
dffull	24	F1,24	4.25968
SSEred	1262069		
Betared	21	Conclude:	Reduced at least as good

Table C.14. Categorical Model REDUCTION 13

Response:	Effort
Summary of Fit	
RSquare	0.823536
RSquare Adj	0.687795
Root Mean Sq Error	221.6883
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	15784.2	0.3212	0.5758	
Sched	1	1	163783	3.3326	0.0794	
AC_H	1	1	385578	7.8456	0.0095	
RV_L	1	1	102665	2.089	0.1603	
TC_H	1	1	287278	5.8454	0.0229	
LE_L	1	1	62414.6	1.27	0.2701	
SL_H	1	1	64014.2	1.3025	0.2642	
RSL_H	1	1	27859.1	0.5669	0.4583	
AT_L	1	1	11499.4	0.234	0.6326	
AT_H	1	1	113904	2.3177	0.14	
achp	1	1	300657	6.1177	0.0202	
rvlp	1	1	108757	2.213	0.1489	
lelp	1	1	16047.6	0.3265	0.5726	remove
atlp	1	1	28551.5	0.581	0.4528	
AC_H*Norm_siz	1	1	197020	4.0089	0.0558	
TC_H*Norm_siz	1	1	845619	17.2064	0.0003	
LE_L*Norm_siz	1	1	6815.19	0.1387	0.7126	
AT_L*Norm_siz	1	1	38964.6	0.7928	0.3814	
AT_H*Norm_siz	1	1	109627	2.2307	0.1473	
RV_L*Norm_siz	1	1	92843.7	1.8892	0.181	

Whole-Model Test				
Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	20	5963287	298164	6.0669
Error	26	1277788	49146	Prob>F
C Total	46	7241074		<.0001

SSEfull	1262069	Test Stat	0.31136
Betafull	21		
dffull	25	F1,25	4.2417
SSEred	1277788		
Betared	20	Conclude:	Reduced at least as good

Table C.15. Categorical Model REDUCTION 14

Response:	Effort
Summary of Fit	
RSquare	0.82132
RSquare Adj	0.695582
Root Mean Sq Error	218.906
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	2190.1	0.0457	0.8323	
Sched	1	1	176246	3.6779	0.0658	
AC_H	1	1	381910	7.9698	0.0088	
RV_L	1	1	114623	2.392	0.1336	
TC_H	1	1	338959	7.0735	0.013	
LE_L	1	1	51089	1.0661	0.311	
SL_H	1	1	51711.2	1.0791	0.3081	
RSL_H	1	1	21465.1	0.4479	0.509	remove
AT_L	1	1	11332.1	0.2365	0.6307	
AT_H	1	1	137144	2.8619	0.1022	
achp	1	1	289136	6.0337	0.0208	
rvlp	1	1	114761	2.3949	0.1334	
atlp	1	1	32877.6	0.6861	0.4148	
AC_H*Norm_siz	1	1	186095	3.8835	0.0591	
TC_H*Norm_siz	1	1	1027815	21.4486	<.0001	
LE_L*Norm_siz	1	1	67702.9	1.4128	0.2449	
AT_L*Norm_siz	1	1	44239.5	0.9232	0.3452	
AT_H*Norm_siz	1	1	118382	2.4704	0.1277	
RV_L*Norm_siz	1	1	97402.3	2.0326	0.1654	

Whole-Model Test				
Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio

Model	19	5947239	313013	6.532
Error	27	1293835	47920	Prob>F
C Total	46	7241074		<.0001

SSEfull	1277788	Test Stat	0.32653
Betafull	20		
dffull	26	F1,26	4.2252
SSEred	1293835		
Betared	19	Conclude:	Reduced at least as good

Table C.16. Categorical Model REDUCTION 15

Response:	Effort
Summary of Fit	
RSquare	0.818356
RSquare Adj	0.701584
Root Mean Sq Error	216.7372
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	1119.4	0.0238	0.8784	
Sched	1	1	165808	3.5297	0.0707	
AC_H	1	1	360594	7.6763	0.0098	
RV_L	1	1	93213.6	1.9843	0.1699	
TC_H	1	1	354091	7.5379	0.0104	
LE_L	1	1	65928.3	1.4035	0.2461	
SL_H	1	1	32358	0.6888	0.4136	
AT_L	1	1	4968.4	0.1058	0.7474	
AT_H	1	1	184271	3.9227	0.0575	
achp	1	1	268893	5.7242	0.0237	
rvlp	1	1	93342.5	1.9871	0.1697	
atlp	1	1	17728.6	0.3774	0.544	remove
AC_H*Norm_siz	1	1	164706	3.5062	0.0716	
TC_H*Norm_siz	1	1	1047236	22.2935	<.0001	
LE_L*Norm_siz	1	1	52676.4	1.1214	0.2987	
AT_L*Norm_siz	1	1	26839	0.5713	0.456	
AT_H*Norm_siz	1	1	117533	2.502	0.1249	
RV_L*Norm_siz	1	1	76072.7	1.6194	0.2136	

Whole-Model Test

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	18	5925774	329210	7.0082
Error	28	1315300	46975	Prob>F
C Total	46	7241074		<.0001

SSEfull	1293835	Test Stat	0.44794
Betafull	19		
dffull	27	F1,27	4.21001
SSEred	1315300		
Betared	18	Conclude:	Reduced at least as good

Table C.17. Categorical Model REDUCTION 16

Response:	Effort
Summary of Fit	
RSquare	0.815907
RSquare Adj	0.707991
Root Mean Sq Error	214.398
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	7653	0.1665	0.6862	
Sched	1	1	150724	3.279	0.0805	
AC_H	1	1	395511	8.6043	0.0065	
RV_L	1	1	75792.3	1.6489	0.2093	
TC_H	1	1	336363	7.3176	0.0113	
LE_L	1	1	52829.3	1.1493	0.2925	
SL_H	1	1	41336.2	0.8993	0.3508	remove
AT_L	1	1	2248.2	0.0489	0.8265	
AT_H	1	1	169328	3.6837	0.0648	
achp	1	1	332020	7.2231	0.0118	
rvlp	1	1	97640.7	2.1242	0.1557	
AC_H*Norm_siz	1	1	190859	4.1521	0.0508	
TC_H*Norm_siz	1	1	1035282	22.5225	<.0001	
LE_L*Norm_siz	1	1	43964.9	0.9565	0.3362	
AT_L*Norm_siz	1	1	52108.5	1.1336	0.2958	
AT_H*Norm_siz	1	1	119351	2.5965	0.1179	
RV_L*Norm_siz	1	1	73586.1	1.6009	0.2159	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	17	5908045	347532	7.5605
Error	29	1333029	45967	Prob>F
C Total	46	7241074		<.0001

SSEfull	1315300	Test Stat	0.3774
Betafull	18		
dffull	28	F1,28	4.19598
SSEred	1333029		
Betared	17	Conclude:	Reduced at least as good

Table C.18. Categorical Model REDUCTION 17

Response:	Effort
Summary of Fit	
RSquare	0.810199
RSquare Adj	0.708971
Root Mean Sq Error	214.0378
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	5821.9	0.1271	0.724	
Sched	1	1	236141	5.1545	0.0305	
AC_H	1	1	360761	7.8748	0.0087	
RV_L	1	1	51962.8	1.1343	0.2954	
TC_H	1	1	328903	7.1794	0.0119	
LE_L	1	1	41503.3	0.9059	0.3488	
AT_L	1	1	29.5	0.0006	0.9799	
AT_H	1	1	162761	3.5528	0.0692	
achp	1	1	314465	6.8642	0.0137	
rvlp	1	1	84979.5	1.855	0.1833	
AC_H*Norm_siz	1	1	176759	3.8583	0.0588	
TC_H*Norm_siz	1	1	1047392	22.8627	<.0001	
LE_L*Norm_siz	1	1	35080.1	0.7657	0.3885	remove
AT_L*Norm_siz	1	1	36378.6	0.7941	0.38	
AT_H*Norm_siz	1	1	97339.6	2.1248	0.1553	
RV_L*Norm_siz	1	1	63026.6	1.3758	0.2501	

Whole-Model Test

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	16	5866709	366669	8.0038
Error	30	1374365	45812	Prob>F
C Total	46	7241074		<.0001

SSEfull	1333029	Test Stat	0.89927
Betafull	17		
dffull	29	F1,29	4.18297
SSEred	1374365		
Betared	16	Conclude:	Reduced at least as good

Table C.19. Categorical Model REDUCTION 18

Response:	Effort
Summary of Fit	
RSquare	0.805354
RSquare Adj	0.711171
Root Mean Sq Error	213.2275
Mean of Response	416.766

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	24730.5	0.5439	0.4664	
Sched	1	1	201659	4.4354	0.0434	
AC_H	1	1	325711	7.1638	0.0118	
RV_L	1	1	48637.5	1.0698	0.309	
TC_H	1	1	297424	6.5417	0.0156	
LE_L	1	1	6996.9	0.1539	0.6975	remove
AT_L	1	1	13211.2	0.2906	0.5937	
AT_H	1	1	133719	2.9411	0.0963	
achp	1	1	288072	6.336	0.0172	
rvlp	1	1	90049.8	1.9806	0.1693	
AC_H*Norm_siz	1	1	163029	3.5857	0.0676	
TC_H*Norm_siz	1	1	1029135	22.6353	<.0001	
AT_L*Norm_siz	1	1	12877.8	0.2832	0.5984	
AT_H*Norm_siz	1	1	62503.5	1.3747	0.2499	
RV_L*Norm_siz	1	1	69162.8	1.5212	0.2267	

Whole-Model Test

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	15	5831629	388775	8.5509
Error	31	1409445	45466	Prob>F
C Total	46	7241074		<.0001

SSEfull	1374365	Test Stat	0.76574
Betafull	16		
dffull	30	F1,30	4.17089
SSEred	1409445		
Betared	15	Conclude:	Reduced at least as good

Table C.20. Categorical Model REDUCTION 19

Response:	Effort
Summary of Fit	
RSquare	0.804388
RSquare Adj	0.718808
Root Mean Sq Error	210.3897
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F
Norm_siz	1	1	20086.9	0.4538	0.5054
Sched	1	1	203129	4.5891	0.0399
AC_H	1	1	323985	7.3194	0.0108
RV_L	1	1	44453.1	1.0043	0.3238
TC_H	1	1	293276	6.6256	0.0149
AT_L	1	1	34709.6	0.7842	0.3825

AT_H	1	1	146721	3.3147	0.078	
achp	1	1	281408	6.3575	0.0169	
rvlp	1	1	83451.4	1.8853	0.1793	
AC_H*Norm_siz	1	1	157672	3.5621	0.0682	
TC_H*Norm_siz	1	1	1023466	23.1219	<.0001	
AT_L*Norm_siz	1	1	8576.2	0.1938	0.6628	remove
AT_H*Norm_siz	1	1	56505.7	1.2766	0.2669	
RV_L*Norm_siz	1	1	62909.2	1.4212	0.242	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	14	5824632	416045	9.3992
Error	32	1416442	44264	Prob>F
C Total	46	7241074		<.0001

SSEfull	1409445	Test Stat	0.15389
Betafull	15		
dffull	31	F1,31	4.15962
SSEred	1416442		
Betared	14	Conclude:	Reduced at least as good

Table C.21. Categorical Model REDUCTION 20

Response:	Effort
Summary of Fit	
RSquare	0.803203
RSquare Adj	0.725678
Root Mean Sq Error	207.8037
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	15707.7	0.3638	0.5506	
Sched	1	1	213492	4.944	0.0331	
AC_H	1	1	340612	7.8878	0.0083	
RV_L	1	1	39089	0.9052	0.3483	
TC_H	1	1	284838	6.5962	0.0149	
AT_L	1	1	194343	4.5005	0.0415	
AT_H	1	1	149655	3.4657	0.0716	
achp	1	1	277816	6.4335	0.0161	
rvlp	1	1	76448.1	1.7704	0.1925	
AC_H*Norm_siz	1	1	151587	3.5104	0.0699	
TC_H*Norm_siz	1	1	1015914	23.5261	<.0001	
AT_H*Norm_siz	1	1	54091.5	1.2526	0.2711	remove
RV_L*Norm_siz	1	1	56203.1	1.3015	0.2621	

Whole-Model Test

Analysis of Variance	DF	Sum of Squares	Mean Square	F Ratio
Source				
Model	13	5816056	447389	10.3605
Error	33	1425018	43182	Prob>F
C Total	46	7241074		<.0001

SSEfull	1416442	Test Stat	0.19375
Betafull	14		
dffull	32	F1,32	4.14909
SSEred	1425018		
Betared	13	Conclude:	Reduced at least as good

Table C.22. Categorical Model REDUCTION 21

Response:	Effort
Summary of Fit	
RSquare	0.795733
RSquare Adj	0.723639
Root Mean Sq Error	208.5743
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	36656.3	0.8426	0.3651	
Sched	1	1	246482	5.6658	0.023	
AC_H	1	1	489521	11.2525	0.002	
RV_L	1	1	101104	2.3241	0.1366	
TC_H	1	1	289664	6.6584	0.0144	
AT_L	1	1	159686	3.6707	0.0638	
AT_H	1	1	99455.2	2.2862	0.1398	remove
achp	1	1	572919	13.1696	0.0009	
rvlp	1	1	175076	4.0244	0.0529	
AC_H*Norm_siz	1	1	415606	9.5535	0.004	
TC_H*Norm_siz	1	1	1115071	25.6319	<.0001	
RV_L*Norm_siz	1	1	138919	3.1933	0.0829	

Whole-Model Test

Analysis of Variance	DF	Sum of Squares	Mean Square	F Ratio
Source				
Model	12	5761964	480164	11.0374
Error	34	1479110	43503	Prob>F
C Total	46	7241074		<.0001

SSEfull	1425018	Test Stat	1.25263
Betafull	13		

dffull	33	F1,33	4.13925
SSEred	1479110		
Betared	12	Conclude:	Reduced at least as good

Table C.23. Categorical Model REDUCTION 22

Response:	Effort
Summary of Fit	
RSquare	0.781998
RSquare Adj	0.713484
Root Mean Sq Error	212.372
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	32299.6	0.7161	0.4032	
Sched	1	1	261176	5.7908	0.0215	
AC_H	1	1	468805	10.3944	0.0027	
RV_L	1	1	100991	2.2392	0.1435	
TC_H	1	1	221426	4.9095	0.0333	
AT_L	1	1	422600	9.3699	0.0042	
achp	1	1	548244	12.1557	0.0013	
rvlp	1	1	159363	3.5334	0.0685	remove
AC_H*Norm_siz	1	1	394953	8.7569	0.0055	
TC_H*Norm_siz	1	1	1015927	22.5252	<.0001	
RV_L*Norm_siz	1	1	126833	2.8121	0.1025	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	11	5662509	514774	11.4136
Error	35	1578565	45102	Prob>F
C Total	46	7241074		<.0001

SSEfull	1479110	Test Stat	2.28616
Betafull	12		
dffull	34	F1,34	4.13002
SSEred	1578565		
Betared	11	Conclude:	Reduced at least as good

Table C.24. Categorical Model REDUCTION 23

Response:	Effort
Summary of Fit	
RSquare	0.75999
RSquare Adj	0.693321
Root Mean Sq Error	219.7175
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Norm_siz	1	1	149439	3.0955	0.087	
Sched	1	1	219952	4.5562	0.0397	remove
AC_H	1	1	376829	7.8058	0.0083	
RV_L	1	1	925.2	0.0192	0.8907	
TC_H	1	1	185053	3.8332	0.058	
AT_L	1	1	518000	10.73	0.0023	
achp	1	1	460583	9.5407	0.0039	
AC_H*Norm_siz	1	1	238745	4.9454	0.0325	
TC_H*Norm_siz	1	1	1173824	24.315	<.0001	
RV_L*Norm_siz	1	1	307756	6.375	0.0161	

Whole-Model Test

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	10	5503147	550315	11.3994
Error	36	1737928	48276	Prob>F
C Total	46	7241074		<.0001

SSEfull	1578565	Test Stat	3.5334
Betafull	11		
dffull	35	F1,35	4.12135
SSEred	1737928		
Betared	10	Conclude:	Reduced at least as good

Table C.25. Categorical Model REDUCTION 24

Response:	Effort
Summary of Fit	
RSquare	0.729615
RSquare Adj	0.663845
Root Mean Sq Error	230.0341
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F
Norm_siz	1	1	117791	2.226	0.1442
AC_H	1	1	219164	4.1418	0.049
RV_L	1	1	31453.7	0.5944	0.4456
TC_H	1	1	55021.1	1.0398	0.3145
AT_L	1	1	591441	11.1771	0.0019
achp	1	1	391915	7.4064	0.0098
AC_H*Norm_siz	1	1	210196	3.9723	0.0537
TC_H*Norm_siz	1	1	956735	18.0804	0.0001
RV_L*Norm_siz	1	1	195895	3.702	0.0621

Whole-Model Test

Analysis of Variance	DF	Sum of Squares	Mean Square	F Ratio
Source				
Model	9	5283195	587022	11.0935
Error	37	1957880	52916	Prob>F
C Total	46	7241074		<.0001

SSEfull	1737928	Test Stat	4.55615
Betafull	10		
dffull	36	F1,36	4.11316
SSEred	1957880		
Betared	9	Conclude:	Cannot reduce

Table C.26. Categorical Model FINAL MODEL

Response: Effort

Summary of Fit

Rsquare	0.75999
RSquare Adj	0.693321
Root Mean Sq Error	219.7175
Mean of Response	416.766
Observations	47

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F
Norm_siz	1	1	1880.1	0.0389	0.8447
Sched	1	1	219952	4.5562	0.0397
AC_H	1	1	376829	7.8058	0.0083
RV_L	1	1	925.2	0.0192	0.8907
TC_H	1	1	185053	3.8332	0.058
AT_L	1	1	518000	10.73	0.0023
Achp	1	1	460583	9.5407	0.0039
AC_H*Norm_siz	1	1	238745	4.9454	0.0325
TC_H*Norm_siz	1	1	1173824	24.315	<.0001
RV_L*Norm_siz	1	1	307756	6.375	0.0161

Whole-Model Test

Analysis of Variance	DF	Sum of Squares	Mean Square	F Ratio
Source				
Model	10	5503147	550315	11.3994
Error	36	1737928	48276	Prob>F
C Total	46	7241074		<.0001

Appendix D. Continuous Model Reductions

Table D.1. Continuous Model FULL MODEL

Response:	Effort
Summary of Fit	
RSquare	0.88634
RSquare Adj	0.55481
Root Mean Sq Error	264.787
Mean of Response	408.3
Observations	48

Effect Test Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
language	1	1	8729.66	0.1245	0.7303	
Size	1	1	55581.7	0.7928	0.3908	
SLOC N	1	1	32.37	0.0005	0.9832	remove
SLOC R	1	1	246.8	0.0035	0.9537	remove
schedule	1	1	143640	2.0487	0.1779	
AC	1	1	14.68	0.0002	0.9887	
RV	1	1	670.69	0.0096	0.9237	
TC	1	1	125334	1.7876	0.206	
PE	1	1	127033	1.8119	0.2032	
LE	1	1	24758.3	0.3531	0.5634	
SL	1	1	4662.75	0.0665	0.8009	
RSL	1	1	86427.1	1.2327	0.2886	
MPE	1	1	161.29	0.0023	0.9625	
ATS	1	1	40897	0.5833	0.4598	
size ac	1	1	369.75	0.0053	0.9433	
size rv	1	1	22367.6	0.319	0.5826	
size tc	1	1	69831.9	0.996	0.338	
size pe	1	1	94873.6	1.3532	0.2673	
size le	1	1	41564.4	0.5928	0.4562	
size sl	1	1	58687	0.837	0.3783	
size rsl	1	1	88828.8	1.267	0.2823	
size mpe	1	1	6883.07	0.0982	0.7594	
size ats	1	1	39453.9	0.5627	0.4676	
size/sched	1	1	1039.45	0.0148	0.9051	
ss^.75	1	1	2778.76	0.0396	0.8455	
size^.75	1	1	40226.3	0.5737	0.4634	
AC*Size	1	1	1872.81	0.0267	0.8729	
RV*Size	1	1	31260.1	0.4459	0.5169	
TC*Size	1	1	50603	0.7217	0.4122	
PE*Size	1	1	75116.7	1.0714	0.321	
LE*Size	1	1	54906.5	0.7831	0.3936	
SL*Size	1	1	68203.7	0.9728	0.3435	
RSL*Size	1	1	95956.4	1.3686	0.2648	

MPE*Size	1	1	24941.9	0.3557	0.562
ATS*Size	1	1	32955.6	0.47	0.506

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	35	6560628	187447	2.6735
Error	12	841343	70112	Prob>F
C Total	47	7401970		0.0358

Table D.2. Continuous Model REDUCTION 1:

Response:	Effort
Summary of Fit	
RSquare	0.88602
RSquare Adj	0.61736
Root Mean Sq Error	245.483
Mean of Response	408.3
Observations	48

Effect Test			Sum of			
Source	Nparm	DF	Squares	F Ratio	Prob>F	
language	1	1	6782.2	0.1125	0.7422	
Size	1	1	146348	2.4285	0.1415	
schedule	1	1	154304	2.5605	0.1319	
AC	1	1	282.37	0.0047	0.9464	
RV	1	1	600.22	0.01	0.9219	
TC	1	1	175523	2.9127	0.11	
PE	1	1	160484	2.6631	0.125	
LE	1	1	78316.2	1.2996	0.2734	
SL	1	1	7838.64	0.1301	0.7237	
RSL	1	1	163031	2.7054	0.1223	
MPE	1	1	196.21	0.0033	0.9553	
ATS	1	1	66374.5	1.1014	0.3117	
size ac	1	1	1765.71	0.0293	0.8665	
size rv	1	1	30241.8	0.5018	0.4903	
size tc	1	1	133045	2.2078	0.1595	
size pe	1	1	122135	2.0267	0.1765	
size le	1	1	143581	2.3826	0.145	
size sl	1	1	135642	2.2509	0.1558	
size rsl	1	1	181841	3.0175	0.1043	
size mpe	1	1	35550.2	0.5899	0.4552	
size ats	1	1	58433.9	0.9697	0.3415	
size/sched	1	1	18238.9	0.3027	0.5909	
ss^.75	1	1	594.77	0.0099	0.9223	remove
size^.75	1	1	101517	1.6846	0.2153	
AC*Size	1	1	5297.83	0.0879	0.7712	
RV*Size	1	1	44083	0.7315	0.4068	
TC*Size	1	1	109905	1.8238	0.1983	
PE*Size	1	1	100359	1.6654	0.2178	

LE*Size	1	1	176269	2.925	0.1093
SL*Size	1	1	166247	2.7587	0.1189
RSL*Size	1	1	201015	3.3357	0.0892
MPE*Size	1	1	110160	1.828	0.1978
ATS*Size	1	1	49766.8	0.8258	0.3789

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	33	6558302	198736	3.2979
Error	14	843668	60262	Prob>F
C Total	47	7401970		0.0103

SSEfull	841343	Test Stat	0.01659
Betafull	35		
dffull	12	F2,12	3.88529
SSEred	843668		
Betared	33	Conclude:	Reduced at least as good

Table D.3. Continuous Model REDUCTION 2:

Response:	Effort
Summary of Fit	
RSquare	0.88594
RSquare Adj	0.64261
Root Mean Sq Error	237.243
Mean of Response	408.3
Observations	48

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
language	1	1	6295.36	0.1118	0.7427	
Size	1	1	150930	2.6816	0.1223	
schedule	1	1	341151	6.0612	0.0264	
AC	1	1	180.35	0.0032	0.9556	
RV	1	1	1226.38	0.0218	0.8846	
TC	1	1	186289	3.3098	0.0889	
PE	1	1	179510	3.1894	0.0943	
LE	1	1	78131.6	1.3882	0.2571	
SL	1	1	13119.9	0.2331	0.6362	
RSL	1	1	162799	2.8925	0.1096	
MPE	1	1	321.34	0.0057	0.9408	
ATS	1	1	68294.5	1.2134	0.288	
size ac	1	1	1460.71	0.026	0.8742	remove
size rv	1	1	34480.2	0.6126	0.446	
size tc	1	1	132984	2.3627	0.1451	
size pe	1	1	137863	2.4494	0.1384	
size le	1	1	154817	2.7506	0.118	
size sl	1	1	148930	2.646	0.1246	

size rsl	1	1	181439	3.2236	0.0928
size mpe	1	1	36109.5	0.6416	0.4357
size ats	1	1	60486.3	1.0747	0.3163
size/sched	1	1	586523	10.4207	0.0056
size^.75	1	1	108938	1.9355	0.1844
AC*Size	1	1	4847.52	0.0861	0.7732
RV*Size	1	1	48787.7	0.8668	0.3666
TC*Size	1	1	109313	1.9422	0.1837
PE*Size	1	1	111750	1.9855	0.1792
LE*Size	1	1	195195	3.468	0.0823
SL*Size	1	1	176267	3.1317	0.0971
RSL*Size	1	1	200490	3.5621	0.0786
MPE*Size	1	1	110666	1.9662	0.1812
ATS*Size	1	1	50661.7	0.9001	0.3578

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	32	6557707	204928	3.641
Error	15	844263	56284	Prob>F
C Total	47	7401970		0.0051

SSEfull	843668	Test Stat	0.00987
Betafull	33		
dffull	14	F1,14	4.60011
SSEred	844263		
Betared	32	Conclude:	Reduced at least as good

Table D.4. Continuous Model REDUCTION 3

Response:	Effort
Summary of Fit	
RSquare	0.88574
RSquare Adj	0.66437
Root Mean Sq Error	229.908
Mean of Response	408.3
Observations	48

Effect Test			Sum of		
Source	Nparm	DF	Squares	F Ratio	Prob>F
language	1	1	7554.04	0.1429	0.7104
Size	1	1	170899	3.2332	0.0911
schedule	1	1	447254	8.4615	0.0103
AC	1	1	4046.81	0.0766	0.7856
RV	1	1	2967.02	0.0561	0.8157
TC	1	1	321188	6.0765	0.0254
PE	1	1	255081	4.8258	0.0431
LE	1	1	89093.3	1.6855	0.2126
SL	1	1	18415.5	0.3484	0.5633

RSL	1	1	233821	4.4236	0.0516	
MPE	1	1	21.85	0.0004	0.984	
ATS	1	1	103236	1.9531	0.1813	
size rv	1	1	55855.4	1.0567	0.3193	
size tc	1	1	249781	4.7255	0.0451	
size pe	1	1	208465	3.9439	0.0645	
size le	1	1	181074	3.4257	0.0827	
size sl	1	1	192464	3.6412	0.0745	
size rsl	1	1	272816	5.1613	0.0372	
size mpe	1	1	39326.5	0.744	0.4011	remove
size ats	1	1	89441.3	1.6921	0.2117	
size/sched	1	1	759654	14.3717	0.0016	
size^.75	1	1	117793	2.2285	0.1549	
AC*Size	1	1	84585.2	1.6002	0.224	
RV*Size	1	1	77326.2	1.4629	0.244	
TC*Size	1	1	196842	3.724	0.0716	
PE*Size	1	1	168481	3.1874	0.0932	
LE*Size	1	1	214364	4.0555	0.0612	
SL*Size	1	1	225500	4.2662	0.0555	
RSL*Size	1	1	305126	5.7726	0.0288	
MPE*Size	1	1	115944	2.1935	0.158	
ATS*Size	1	1	75263.7	1.4239	0.2502	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	31	6556247	211492	4.0012
Error	16	845724	52858	Prob>F
C Total	47	7401970		0.0025

SSEfull	844263	Test Stat	0.02595
Betafull	32		
dffull	15	F1,15	4.54307
SSEred	845724		
Betared	31	Conclude:	Reduced at least as good

Table D.5. Continuous Model REDUCTION 4:

Response:	Effort
Summary of Fit	
RSquare	0.88043
RSquare Adj	0.66943
Root Mean Sq Error	228.171
Mean of Response	408.3
Observations	48

Effect Test			Sum of		
Source	Nparm	DF	Squares	F Ratio	Prob>F
language	1	1	17893	0.3437	0.5654

Size	1	1	134104	2.5759	0.1269
schedule	1	1	409769	7.8708	0.0122
AC	1	1	3195.5	0.0614	0.8073
RV	1	1	884.57	0.017	0.8978
TC	1	1	282582	5.4278	0.0324
PE	1	1	226091	4.3427	0.0526
LE	1	1	54621.3	1.0492	0.3201
SL	1	1	20927.9	0.402	0.5345
RSL	1	1	200164	3.8447	0.0665
MPE	1	1	86872.6	1.6686	0.2137
ATS	1	1	65628.6	1.2606	0.2771
size rv	1	1	37546.6	0.7212	0.4076r
size tc	1	1	214045	4.1114	0.0586
size pe	1	1	184679	3.5473	0.0769
size le	1	1	186174	3.576	0.0758
size sl	1	1	174648	3.3546	0.0846
size rsl	1	1	251577	4.8323	0.0421
size ats	1	1	51752.6	0.9941	0.3327
size/sched	1	1	754687	14.496	0.0014
size^.75	1	1	87607.7	1.6828	0.2119
AC*Size	1	1	75284.7	1.4461	0.2456
RV*Size	1	1	55581.2	1.0676	0.316
TC*Size	1	1	167100	3.2096	0.091
PE*Size	1	1	149169	2.8652	0.1088
LE*Size	1	1	206387	3.9643	0.0628
SL*Size	1	1	203198	3.903	0.0647
RSL*Size	1	1	296007	5.6857	0.029
MPE*Size	1	1	376572	7.2332	0.0155
ATS*Size	1	1	38542.1	0.7403	0.4015

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	30	6516920	217231	4.1726
Error	17	885050	52062	Prob>F
C Total	47	7401970		0.0016

SSEfull	845724	Test Stat	0.74401
Betafull	31		
dffull	16	F1,16	4.494
SSEred	885050		
Betared	30	Conclude:	Reduced at least as good

Table D.6. Continuous Model REDUCTION 5:

Response:	Effort
Summary of Fit	
RSquare	0.87536
RSquare Adj	0.67455

Root Mean Sq Error	226.397
Mean of Response	408.3
Observations	48

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
language	1	1	17573.6	0.3429	0.5655	remove
Size	1	1	97171.6	1.8958	0.1854	
schedule	1	1	372484	7.2672	0.0148	
AC	1	1	8370.2	0.1633	0.6909	
RV	1	1	63655.2	1.2419	0.2798	
TC	1	1	271198	5.2911	0.0336	
PE	1	1	209628	4.0899	0.0583	
LE	1	1	59789.1	1.1665	0.2944	
SL	1	1	6211.04	0.1212	0.7318	
RSL	1	1	219442	4.2814	0.0532	
MPE	1	1	96937.9	1.8913	0.1859	
ATS	1	1	92140.2	1.7977	0.1967	
size tc	1	1	279357	5.4503	0.0313	
size pe	1	1	150417	2.9347	0.1039	
size le	1	1	291479	5.6868	0.0283	
size sl	1	1	137127	2.6754	0.1193	
size rsl	1	1	314859	6.1429	0.0233	
size ats	1	1	112800	2.2007	0.1552	
size/sched	1	1	718183	14.0119	0.0015	
size^.75	1	1	50680.2	0.9888	0.3332	
AC*Size	1	1	79740.3	1.5557	0.2283	
RV*Size	1	1	210713	4.111	0.0577	
TC*Size	1	1	189942	3.7058	0.0702	
PE*Size	1	1	113068	2.206	0.1548	
LE*Size	1	1	319532	6.2341	0.0225	
SL*Size	1	1	165985	3.2384	0.0887	
RSL*Size	1	1	375603	7.3281	0.0144	
MPE*Size	1	1	349610	6.8209	0.0177	
ATS*Size	1	1	101681	1.9838	0.176	

Whole-Model Test

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	29	6479373	223427	4.3591
Error	18	922597	51255	Prob>F
C Total	47	7401970		0.0009

SSEfull	885050	Test Stat	0.72119
Betafull	30		
dffull	17	F1,17	4.45132
SSEred	922597		
Betared	29	Conclude:	Reduced at least as good

Table D.7. Continuous Model REDUCTION 6:

Response:	Effort
Summary of Fit	
Rsquare	0.87298
RSquare Adj	0.6858
Root Mean Sq Error	222.447
Mean of Response	408.3
Observations	48

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Size	1	1	79928.5	1.6153	0.2191	
schedule	1	1	359603	7.2672	0.0143	
AC	1	1	6370.75	0.1287	0.7237	
RV	1	1	66067.7	1.3352	0.2622	
TC	1	1	259025	5.2347	0.0338	
PE	1	1	194627	3.9332	0.062	
LE	1	1	73748.1	1.4904	0.2371	
SL	1	1	1696.15	0.0343	0.8551	
RSL	1	1	201953	4.0813	0.0577	
MPE	1	1	79386.4	1.6043	0.2206	
ATS	1	1	77189.7	1.5599	0.2268	
size tc	1	1	277286	5.6037	0.0287	
size pe	1	1	133093	2.6897	0.1175	
size le	1	1	274448	5.5463	0.0294	
size sl	1	1	119840	2.4219	0.1362	
size rsl	1	1	298257	6.0275	0.0239	
size ats	1	1	97231.9	1.965	0.1771	
size/sched	1	1	725832	14.6684	0.0011	
size^.75	1	1	37062	0.749	0.3976	remove
AC*Size	1	1	71347.6	1.4419	0.2446	
RV*Size	1	1	200358	4.0491	0.0586	
TC*Size	1	1	216632	4.3779	0.0501	
PE*Size	1	1	97269.1	1.9657	0.177	
LE*Size	1	1	311166	6.2884	0.0214	
SL*Size	1	1	149854	3.0284	0.098	
RSL*Size	1	1	359778	7.2708	0.0143	
MPE*Size	1	1	356821	7.211	0.0146	
ATS*Size	1	1	88416.7	1.7868	0.1971	

Whole-Model Test

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	28	6461800	230779	4.6638
Error	19	940171	49483	Prob>F
C Total	47	7401970		0.0005

SSEfull	922597	Test Stat	0.34286
Betafull	29		
dffull	18	F1,18	4.41386
SSEred	940171		
Betared	28	Conclude:	Reduced at least as good

Table D.8. Continuous Model REDUCTION 7:

Response:	Effort
Summary of Fit	
RSquare	0.86798
RSquare Adj	0.68975
Root Mean Sq Error	221.047
Mean of Response	408.3
Observations	48

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Size	1	1	229329	4.6934	0.0425	
schedule	1	1	324851	6.6484	0.0179	
AC	1	1	10.34	0.0002	0.9885	
RV	1	1	40180.6	0.8223	0.3753	
TC	1	1	225715	4.6195	0.044	
PE	1	1	172277	3.5258	0.0751	
LE	1	1	102986	2.1077	0.1621	
SL	1	1	29910	0.6121	0.4431	
RSL	1	1	177391	3.6305	0.0712	
MPE	1	1	44900.6	0.9189	0.3492	
ATS	1	1	71721.4	1.4678	0.2398	
size tc	1	1	252240	5.1623	0.0343	
size pe	1	1	96701	1.9791	0.1748	
size le	1	1	259343	5.3077	0.0321	
size sl	1	1	123612	2.5298	0.1274	
size rsl	1	1	339846	6.9553	0.0158	
size ats	1	1	87014.7	1.7808	0.197	
size/sched	1	1	689224	14.1056	0.0012	
AC*Size	1	1	45615.8	0.9336	0.3455	remove
RV*Size	1	1	163442	3.345	0.0824	
TC*Size	1	1	199908	4.0913	0.0567	
PE*Size	1	1	60414.4	1.2364	0.2793	
LE*Size	1	1	283999	5.8123	0.0257	
SL*Size	1	1	180735	3.6989	0.0688	
RSL*Size	1	1	417125	8.5369	0.0084	
MPE*Size	1	1	349409	7.151	0.0146	
ATS*Size	1	1	79806.4	1.6333	0.2159	

Whole-Model Test

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	27	6424738	237953	4.8699

Error	20	977233	48862	Prob>F
C Total	47	7401970		0.0003

SSEfull	940171	Test Stat	0.74899
Betafull	28		
dffull	19	F1,19	4.38075
SSEred	977233		
Betared	27	Conclude:	Reduced at least as good

Table D.9. Continuous Model REDUCTION 8:

Response:	Effort
Summary of Fit	
RSquare	0.86181
RSquare Adj	0.69073
Root Mean Sq Error	220.697
Mean of Response	408.3
Observations	48

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Size	1	1	184783	3.7938	0.0649	
schedule	1	1	284122	5.8333	0.0249	
AC	1	1	200709	4.1207	0.0552	
RV	1	1	53353.8	1.0954	0.3072	
TC	1	1	181441	3.7252	0.0672	
PE	1	1	157778	3.2393	0.0863	
LE	1	1	202109	4.1495	0.0544	
SL	1	1	11596.5	0.2381	0.6306	
RSL	1	1	150683	3.0937	0.0932	
MPE	1	1	7867.58	0.1615	0.6918	
ATS	1	1	37315	0.7661	0.3913	
size tc	1	1	251358	5.1606	0.0337	
size pe	1	1	93154.2	1.9125	0.1812	
size le	1	1	236282	4.8511	0.0389	
size sl	1	1	139648	2.8671	0.1052	
size rsl	1	1	297994	6.1181	0.022	
size ats	1	1	52881.4	1.0857	0.3093	remove
size/sched	1	1	710630	14.5899	0.001	
RV*Size	1	1	167625	3.4415	0.0777	
TC*Size	1	1	221002	4.5374	0.0452	
PE*Size	1	1	59026.4	1.2119	0.2834	
LE*Size	1	1	247574	5.0829	0.035	
SL*Size	1	1	184487	3.7877	0.0651	
RSL*Size	1	1	372577	7.6493	0.0116	
MPE*Size	1	1	343947	7.0615	0.0147	
ATS*Size	1	1	48932.2	1.0046	0.3276	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	26	6379122	245351	5.0373
Error	21	1022848	48707	Prob>F
C Total	47	7401970		0.0002

SSEfull	977233	Test Stat	0.93357
Betafull	27		
dffull	20	F1,20	4.35125
SSEred	1022848		
Betared	26	Conclude:	Reduced at least as good

Table D.10. Continuous Model REDUCTION 9:

Response:	Effort
Summary of Fit	
RSquare	0.85467
RSquare Adj	0.68952
Root Mean Sq Error	221.126
Mean of Response	408.3
Observations	48

Effect Test			Sum of		
Source	Nparm	DF	Squares	F Ratio	Prob>F
Size	1	1	134366	2.748	0.1116
schedule	1	1	231428	4.733	0.0406
AC	1	1	211097	4.3172	0.0496
RV	1	1	90317.3	1.8471	0.1879
TC	1	1	129558	2.6496	0.1178
PE	1	1	140960	2.8828	0.1036
LE	1	1	197647	4.0421	0.0568
SL	1	1	3721.77	0.0761	0.7852
RSL	1	1	98408	2.0126	0.17
MPE	1	1	255.84	0.0052	0.943
ATS	1	1	112.26	0.0023	0.9622
size tc	1	1	231659	4.7377	0.0405
size pe	1	1	87157.4	1.7825	0.1955
size le	1	1	185174	3.787	0.0645
size sl	1	1	160854	3.2897	0.0834
size rsl	1	1	321680	6.5787	0.0177
size/sched	1	1	657853	13.4539	0.0014
RV*Size	1	1	222491	4.5502	0.0443
TC*Size	1	1	230445	4.7129	0.041
PE*Size	1	1	58311.6	1.1925	0.2866
LE*Size	1	1	194705	3.9819	0.0585
SL*Size	1	1	200532	4.1011	0.0552
RSL*Size	1	1	526951	10.7768	0.0034

MPE*Size	1	1	292815	5.9884	0.0228
ATS*Size	1	1	734.34	0.015	0.9036remove

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	25	6326241	253050	5.1752
Error	22	1075730	48897	Prob>F
C Total	47	7401970		0.0001

SSEfull	1022848	Test Stat	1.0857
Betafull	26		
dffull	21	F1,21	4.32479
SSEred	1075730		
Betared	25	Conclude:	Reduced at least as good

Table D.11. Continuous Model REDUCTION 10:

Response:	Effort
Summary of Fit	
RSquare	0.85457
RSquare Adj	0.70282
Root Mean Sq Error	216.34
Mean of Response	408.3
Observations	48

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F
Size	1	1	149899	3.2028	0.0867
schedule	1	1	259786	5.5507	0.0274
AC	1	1	233745	4.9943	0.0354
RV	1	1	89730.7	1.9172	0.1795
TC	1	1	155039	3.3126	0.0818
PE	1	1	150901	3.2242	0.0857
LE	1	1	199231	4.2568	0.0506
SL	1	1	5007.49	0.107	0.7466
RSL	1	1	98773	2.1104	0.1598
MPE	1	1	302.48	0.0065	0.9366
ATS	1	1	7353.84	0.1571	0.6955remove
size tc	1	1	237102	5.066	0.0343
size pe	1	1	94795.7	2.0254	0.1681
size le	1	1	193395	4.1321	0.0538
size sl	1	1	243831	5.2098	0.032
size rsl	1	1	328786	7.0249	0.0143
size/sched	1	1	665931	14.2284	0.001
RV*Size	1	1	225007	4.8075	0.0387
TC*Size	1	1	232449	4.9666	0.0359
PE*Size	1	1	63207.1	1.3505	0.2571
LE*Size	1	1	203202	4.3417	0.0485

SL*Size	1	1	336356	7.1867	0.0133
RSL*Size	1	1	561104	11.9887	0.0021
MPE*Size	1	1	292570	6.2511	0.02

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	24	6325506	263563	5.6313
Error	23	1076464	46803	Prob>F
C Total	47	7401970		<.0001

SSEfull	1075730	Test Stat	0.01502
Betafull	25		
dffull	22	F1,22	4.30094
SSEred	1076464		
Betared	24	Conclude:	Reduced at least as good

Table D.12. Continuous Model REDUCTION 11:

Response:	Effort
Summary of Fit	
RSquare	0.85358
RSquare Adj	0.71326
Root Mean Sq Error	212.507
Mean of Response	408.3
Observations	48

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Size	1	1	148369	3.2855	0.0824	
schedule	1	1	303261	6.7154	0.016	
AC	1	1	300548	6.6553	0.0164	
RV	1	1	85816.7	1.9003	0.1808	
TC	1	1	220309	4.8785	0.037	
PE	1	1	153285	3.3943	0.0778	
LE	1	1	216740	4.7995	0.0384	
SL	1	1	3191.7	0.0707	0.7926	
RSL	1	1	145292	3.2173	0.0855	
MPE	1	1	1615.35	0.0358	0.8516	
size tc	1	1	292128	6.4689	0.0178	
size pe	1	1	92175.2	2.0411	0.166	remove
size le	1	1	196880	4.3597	0.0476	
size sl	1	1	268652	5.949	0.0225	
size rsl	1	1	437890	9.6966	0.0047	
size/sched	1	1	792234	17.5432	0.0003	
RV*Size	1	1	218229	4.8324	0.0378	
TC*Size	1	1	266949	5.9113	0.0229	
PE*Size	1	1	59414.3	1.3157	0.2627	
LE*Size	1	1	204968	4.5388	0.0436	

SL*Size	1	1	354681	7.854	0.0099
RSL*Size	1	1	687394	15.2216	0.0007
MPE*Size	1	1	287798	6.373	0.0186

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	23	6318152	274702	6.083
Error	24	1083818	45159	Prob>F
C Total	47	7401970		<.0001

SSEfull	1076464	Test Stat	0.15713
Betafull	24		
dffull	23	F1,23	4.27934
SSEred	1083818		
Betared	23	Conclude:	Reduced at least as good

Table D.13. Continuous Model REDUCTION 12:

Response:	Effort
Summary of Fit	
RSquare	0.84112
RSquare Adj	0.70131
Root Mean Sq Error	216.886
Mean of Response	408.3
Observations	48

Effect Test			Sum of			
Source	Nparm	DF	Squares	F Ratio	Prob>F	
Size	1	1	90843.9	1.9312	0.1769	
schedule	1	1	311118	6.6139	0.0164	
AC	1	1	254115	5.4021	0.0285	
RV	1	1	10461.1	0.2224	0.6413	
TC	1	1	184777	3.9281	0.0586	
PE	1	1	82782.1	1.7598	0.1966	
LE	1	1	126397	2.687	0.1137	
SL	1	1	303.14	0.0064	0.9367	
RSL	1	1	156256	3.3218	0.0804	
MPE	1	1	1808.51	0.0384	0.8461	
size tc	1	1	200155	4.255	0.0497	
size le	1	1	111230	2.3646	0.1367	remove
size sl	1	1	266382	5.6629	0.0253	
size rsl	1	1	359193	7.6359	0.0106	
size/sched	1	1	728645	15.49	0.0006	
RV*Size	1	1	128485	2.7314	0.1109	
TC*Size	1	1	174786	3.7157	0.0653	
PE*Size	1	1	223446	4.7502	0.0389	
LE*Size	1	1	127931	2.7196	0.1116	
SL*Size	1	1	326600	6.9431	0.0142	

RSL*Size	1	1	598694	12.7274	0.0015
MPE*Size	1	1	223510	4.7515	0.0389

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	22	6225977	282999	6.0162
Error	25	1175993	47040	Prob>F
C Total	47	7401970		<.0001

SSEfull	1083818	Test Stat	2.04112
Betafull	23		
dffull	24	F1,24	4.25968
SSEred	1175993		
Betared	22	Conclude:	Reduced at least as good

Table D.14. Continuous Model REDUCTION 13:

Response:	Effort
Summary of Fit	
RSquare	0.8261
RSquare Adj	0.68564
Root Mean Sq Error	222.505
Mean of Response	408.3
Observations	48

Effect Test			Sum of			
Source	Nparm	DF	Squares	F Ratio	Prob>F	
Size	1	1	112811	2.2786	0.1432	
schedule	1	1	298029	6.0198	0.0212	
AC	1	1	216512	4.3732	0.0464	
RV	1	1	60178	1.2155	0.2803	
TC	1	1	180743	3.6507	0.0671	
PE	1	1	43870.5	0.8861	0.3552	
LE	1	1	30782.7	0.6218	0.4375	
SL	1	1	460.26	0.0093	0.9239	
RSL	1	1	82321.9	1.6628	0.2086	
MPE	1	1	1996.85	0.0403	0.8424	
size tc	1	1	318770	6.4387	0.0175	
size sl	1	1	270290	5.4595	0.0274	
size rsl	1	1	248666	5.0227	0.0338	
size/sched	1	1	732692	14.7993	0.0007	
RV*Size	1	1	390534	7.8882	0.0093	
TC*Size	1	1	351342	7.0966	0.0131	
PE*Size	1	1	134609	2.7189	0.1112	
LE*Size	1	1	24628.4	0.4975	0.4869	remove
SL*Size	1	1	342181	6.9115	0.0142	
RSL*Size	1	1	494685	9.9919	0.004	
MPE*Size	1	1	218020	4.4037	0.0457	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	21	6114747	291178	5.8814
Error	26	1287223	49509	Prob>F
C Total	47	7401970		<.0001

SSEfull	1175993	Test Stat	2.3646
Betafull	22		
dffull	25	F1,25	4.2417
SSEred	1287223		
Betared	21	Conclude:	Reduced at least as good

Table D.15. Continuous Model REDUCTION 14:

Response:	Effort
Summary of Fit	
RSquare	0.82277
RSquare Adj	0.69149
Root Mean Sq Error	220.425
Mean of Response	408.3
Observations	48

Effect Test			Sum of			
Source	Nparm	DF	Squares	F Ratio	Prob>F	
Size	1	1	89937.1	1.851	0.1849	
schedule	1	1	278523	5.7325	0.0239	
AC	1	1	213231	4.3886	0.0457	
RV	1	1	99929.3	2.0567	0.163	
TC	1	1	161539	3.3247	0.0793	
PE	1	1	84388.1	1.7368	0.1986	
LE	1	1	253325	5.2138	0.0305	
SL	1	1	1190.79	0.0245	0.8768	
RSL	1	1	116198	2.3915	0.1336	
MPE	1	1	21695.8	0.4465	0.5097	
size tc	1	1	331321	6.8191	0.0145	
size sl	1	1	246310	5.0695	0.0327	
size rsl	1	1	306737	6.3131	0.0183	
size/sched	1	1	709143	14.5953	0.0007	
RV*Size	1	1	449584	9.2532	0.0052	
TC*Size	1	1	401060	8.2545	0.0078	
PE*Size	1	1	178882	3.6817	0.0656	remove
SL*Size	1	1	327897	6.7486	0.015	
RSL*Size	1	1	528960	10.8868	0.0027	
MPE*Size	1	1	228186	4.6964	0.0392	

Whole-Model Test

Analysis of Variance		Sum of	Mean	
Source	DF	Squares	Square	F Ratio
Model	20	6090119	304506	6.2672
Error	27	1311852	48587	Prob>F
C Total	47	7401970		<.0001

SSEfull	1287223	Test Stat	0.49746
Betafull	21		
dffull	26	F1,26	4.2252
SSEred	1311852		
Betared	20	Conclude:	Reduced at least as good

Table D.16. Continuous Model REDUCTION 15:

Response:	Effort
Summary of Fit	
RSquare	0.7986
RSquare Adj	0.66194
Root Mean Sq Error	230.739
Mean of Response	408.3
Observations	48

Effect Test			Sum of			
Source	Nparm	DF	Squares	F Ratio	Prob>F	
Size	1	1	79380.2	1.491	0.2322	
schedule	1	1	178827	3.3589	0.0775	
AC	1	1	290340	5.4534	0.0269	
RV	1	1	32402.2	0.6086	0.4419	
TC	1	1	173819	3.2648	0.0815	
PE	1	1	5384.73	0.1011	0.7528	remove
LE	1	1	262466	4.9298	0.0347	
SL	1	1	5886.43	0.1106	0.742	
RSL	1	1	154734	2.9063	0.0993	
MPE	1	1	72241.4	1.3569	0.2539	
size tc	1	1	439226	8.2498	0.0077	
size sl	1	1	430585	8.0876	0.0082	
size rsl	1	1	367133	6.8957	0.0138	
size/sched	1	1	536282	10.0728	0.0036	
RV*Size	1	1	351186	6.5962	0.0158	
TC*Size	1	1	572734	10.7575	0.0028	
SL*Size	1	1	519274	9.7534	0.0041	
RSL*Size	1	1	589057	11.0641	0.0025	
MPE*Size	1	1	153318	2.8797	0.1008	

Whole-Model Test

Analysis of Variance Source	DF	Sum of Squares	Mean Square	F Ratio
Model	19	5911237	311118	5.8436
Error	28	1490734	53240	Prob>F
C Total	47	7401970		<.0001

SSEfull	1311852	Test Stat	3.68168
Betafull	20		
dffull	27	F1,27	4.21001
SSEred	1490734		
Betared	19	Conclude:	Reduced at least as good

Table D.17. Continuous Model REDUCTION 16:

Response:	Effort
Summary of Fit	
RSquare	0.79788
RSquare Adj	0.67242
Root Mean Sq Error	227.135
Mean of Response	408.3
Observations	48

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Size	1	1	73997.6	1.4343	0.2408	
schedule	1	1	173971	3.3722	0.0766	
AC	1	1	287810	5.5788	0.0251	
RV	1	1	51367.8	0.9957	0.3266	
TC	1	1	173237	3.3579	0.0772	
LE	1	1	265718	5.1505	0.0309	
SL	1	1	9308.6	0.1804	0.6741	
RSL	1	1	153433	2.9741	0.0953	
MPE	1	1	66999.6	1.2987	0.2638	
size tc	1	1	447463	8.6734	0.0063	
size sl	1	1	441481	8.5574	0.0066	
size rsl	1	1	372109	7.2128	0.0118	
size/sched	1	1	545947	10.5824	0.0029	
RV*Size	1	1	369797	7.1679	0.0121	
TC*Size	1	1	589949	11.4353	0.0021	
SL*Size	1	1	520804	10.095	0.0035	
RSL*Size	1	1	593870	11.5113	0.002	
MPE*Size	1	1	153097	2.9676	0.0956	remove

Whole-Model Test

Analysis of Variance Source	DF	Sum of Squares	Mean Square	F Ratio
Model	18	5905852	328103	6.3598

Error	29	1496118	51590	Prob>F
C Total	47	7401970		<.0001

SSEfull	1490734	Test Stat	0.10114
Betafull	19		
dffull	28	F1,28	4.19598
SSEred	1496118		
Betared	18	Conclude:	Reduced at least as good

Table D.18. Continuous Model REDUCTION 17:

Response:	Effort
Summary of Fit	
RSquare	0.77719
RSquare Adj	0.65094
Root Mean Sq Error	234.465
Mean of Response	408.3
Observations	48

Effect Test

Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Size	1	1	2681.12	0.0488	0.8267	
schedule	1	1	74969.3	1.3637	0.2521	
AC	1	1	275977	5.0201	0.0326	
RV	1	1	13259.4	0.2412	0.6269	
TC	1	1	119699	2.1774	0.1505	
LE	1	1	399490	7.2669	0.0114	
SL	1	1	63398.3	1.1532	0.2914	
RSL	1	1	183747	3.3424	0.0775	
MPE	1	1	628995	11.4417	0.002	
size tc	1	1	446772	8.127	0.0078	
size sl	1	1	370881	6.7465	0.0144	
size rsl	1	1	393870	7.1647	0.0119	
size/sched	1	1	460852	8.3831	0.007	
RV*Size	1	1	219275	3.9887	0.0549	remove
TC*Size	1	1	600320	10.9201	0.0025	
SL*Size	1	1	384002	6.9852	0.0129	
RSL*Size	1	1	559626	10.1799	0.0033	

Whole-Model Test
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	17	5752755	338397	6.1556
Error	30	1649215	54974	Prob>F
C Total	47	7401970		<.0001

SSEfull	1496118	Test Stat	2.96755
Betafull	18		

dffull	29	F1,29	4.18297
SSEred	1649215		
Betared	17	Conclude:	Reduced at least as good

Table D.19. Continuous Model REDUCTION 18 (Final Model):

Response:	Effort
Summary of Fit	
RSquare	0.74757
RSquare Adj	0.61728
Root Mean Sq Error	245.507
Mean of Response	408.3
Observations	48

Effect Test Source	Nparm	DF	Sum of Squares	F Ratio	Prob>F	
Size	1	1	28254.3	0.4688	0.4986	
schedule	1	1	191769	3.1816	0.0843	
AC	1	1	319078	5.2938	0.0283	
RV	1	1	259527	4.3058	0.0464	remove
TC	1	1	367710	6.1006	0.0192	
LE	1	1	303311	5.0322	0.0322	
SL	1	1	105195	1.7453	0.1961	
RSL	1	1	510375	8.4676	0.0066	
MPE	1	1	470802	7.811	0.0088	
size tc	1	1	535875	8.8907	0.0055	
size sl	1	1	571404	9.4801	0.0043	
size rsl	1	1	433843	7.1979	0.0116	
size/sched	1	1	425029	7.0516	0.0124	
TC*Size	1	1	645251	10.7053	0.0026	
SL*Size	1	1	591909	9.8203	0.0038	
RSL*Size	1	1	479071	7.9482	0.0083	

Whole-Model Test

Analysis of Variance Source	DF	Sum of Squares	Mean Square	F Ratio
Model	16	5533481	345843	5.7379
Error	31	1868490	60274	Prob>F
C Total	47	7401970		<.0001

SSEfull	1649215	Test Stat	3.98871
Betafull	17		
dffull	30	F1,30	4.17089
SSEred	1868490		
Betared	16	Conclude:	Reduced at least as good

Appendix E. Correlation Matrices

Table E.1. Correlations of the Categorical Model

Variable	Norm_siz	SLOC_N	SLOC_R	Sched	AC_H	RV_L	RV_H
Norm_siz	1						
SLOC_N	0.8849	1					
SLOC_R	0.3731	-0.0949	1				
Sched	-0.1785	-0.1056	-0.1811	1			
AC_H	-0.1963	-0.1558	-0.0793	0.3117	1		
RV_L	0.0665	0.1133	-0.0352	0.0905	0.2374	1	
RV_H	-0.1192	-0.0633	-0.1161	0.135	-0.0485	-0.4445	1
TC_H	-0.0574	-0.0547	0.0134	0.2982	0.0512	0.3926	-0.0713
PE_L	-0.2185	-0.1757	-0.1412	0.3285	-0.0999	0.501	-0.3014
PE_H	0.3091	0.3828	-0.0987	-0.0522	-0.1434	-0.0783	-0.0431
LE_L	-0.2793	-0.1977	-0.1751	-0.1928	0.276	0.0512	0.4761
LE_H	0.1008	-0.0074	0.2062	0.2629	-0.0427	-0.1642	-0.3979
SL_H	0.0138	-0.042	0.1485	0.4648	0.4298	-0.1315	0.2003
RSL_H	0.0597	0.1535	-0.1678	-0.4356	-0.3322	0.3741	-0.0609
MPE_L	-0.1365	-0.09	-0.1008	0.1199	0.1535	-0.1827	0.5053
MPE_H	-0.0452	-0.1494	0.1889	0.1593	0.4592	0.199	-0.5325
AT_L	0.2148	0.1627	0.0539	-0.0379	-0.0996	-0.0682	-0.1733
AT_H	0.0384	0.0016	0.1168	0.1569	0.114	0.3518	-0.4257
language	0.1215	-0.0646	0.4254	-0.115	0.2097	0.3036	-0.1374
size^0.75	0.9907	0.8753	0.3707	-0.202	-0.1972	0.0459	-0.1121
achp	0.4434	0.3949	0.2048	0.0441	0.5975	0.1724	-0.0977
rvlp	0.7353	0.7134	0.1851	-0.0655	0.0043	0.5843	-0.2597
tchp	0.5546	0.4938	0.2449	-0.0217	-0.0784	0.2932	-0.0218
pelp	0.3431	0.3987	-0.0947	0.055	-0.2935	0.3814	-0.2061
lelp	-0.0715	0.005	-0.1483	-0.12	0.2274	0.017	0.4186
atlp	0.5764	0.5933	0.0062	-0.1195	-0.2257	0.0814	-0.1301
athp	0.5108	0.3332	0.4581	-0.0413	0.079	0.1923	-0.2949

Variable	TC_H	PE_L	PE_H	LE_L	LE_H	SL_H	RSL_H
TC_H	1						
PE_L	0.3842	1					
PE_H	-0.3423	-0.4096	1				
LE_L	0.045	-0.1936	-0.3215	1			
LE_H	-0.199	0.0118	0.3735	-0.8607	1		
SL_H	-0.0426	-0.2072	-0.0783	-0.1412	0.2898	1	
RSL_H	0.3373	0.0814	-0.2874	0.3322	-0.5573	-0.5406	1
MPE_L	-0.2753	-0.0074	-0.0971	0.6059	-0.5109	0.0463	-0.2504
MPE_H	0.2551	-0.0806	-0.0654	-0.3588	0.4315	0.2898	-0.1331

AT_L	0.1827	0.109	0.0412	-0.1535	0.0864	-0.2972	0.0171
AT_H	0.081	0.0918	-0.1602	-0.401	0.3478	0.3518	0.0925
language	-0.044	0.1604	0.0626	0.0097	0.0569	-0.0697	-0.2712
size^0.75	-0.0901	-0.2615	0.35	-0.2882	0.1277	0.0156	0.0562
achp	-0.0614	-0.3689	0.2297	-0.0399	0.1678	0.396	-0.2434
rvlp	0.184	0.0932	0.1764	-0.1666	-0.0516	-0.0303	0.2754
tchp	0.645	0.1586	-0.1964	-0.002	-0.2614	-0.0823	0.3253
pelp	0.2909	0.6839	-0.2801	-0.2211	-0.1169	-0.268	0.3275
lelp	0.1616	-0.2537	-0.2585	0.8042	-0.6921	-0.0717	0.3458
atlp	0.177	0.1006	0.0794	-0.111	-0.1029	-0.2908	0.2006
athp	-0.1732	-0.2006	0.1217	-0.3605	0.3396	0.3491	-0.059

Variable	MPE_L	MPE_H	AT_L	AT_H	language	size^0.75	achp
MPE_L	1						
MPE_H	-0.6304	1					
AT_L	-0.0545	0.0864	1				
AT_H	-0.504	0.5284	-0.504	1			
language	0.2621	-0.0316	0.2353	-0.1866	1		
size^0.75	-0.153	-0.0332	0.1979	0.0474	0.1063	1	
achp	-0.1015	0.4276	-0.0543	0.2149	0.2356	0.4898	1
rvlp	-0.2447	0.0732	0.1962	0.1517	0.2191	0.7074	0.4342
tchp	-0.1312	-0.0344	0.3198	-0.1028	0.1336	0.5127	0.1629
pelp	-0.1107	-0.2222	0.2838	-0.0156	0.0029	0.2975	-0.2471
lelp	0.3987	-0.2132	-0.0159	-0.3363	-0.1435	-0.0444	0.1674
atlp	0.0189	-0.1464	0.7506	-0.3783	0.1706	0.525	-0.097
athp	-0.3491	0.3713	-0.3491	0.6928	0.0472	0.544	0.6196

Variable	rvlp	tchp	pelp	lelp	atlp	athp
rvlp	1					
tchp	0.6614	1				
pelp	0.5337	0.5652	1			
lelp	-0.0505	0.1733	-0.1864	1		
atlp	0.5084	0.5957	0.5702	-0.005	1	
athp	0.4237	0.0213	-0.0938	-0.2693	-0.2621	1

Table E.2. Correlations of the Continuous Model

Variable	language	Size	Effort	SLOC N	SLOC R	schedule	AC
language	1						
Size	0.1271	1					
Effort	0.0029	0.4857	1				
SLOC N	-0.0591	0.8855	0.4575	1			
SLOC R	0.4271	0.3752	0.1846	-0.0914	1		
schedule	-0.1024	-0.1615	0.0556	-0.0927	-0.172	1	
AC	0.2408	0.0733	0.0729	0.1163	-0.0537	0.0208	1
RV	-0.3415	-0.0124	0.0611	0.0738	-0.1692	-0.2948	-0.2398
TC	-0.0522	-0.044	0.1166	-0.0277	-0.0011	0.2428	-0.1075
PE	0.2033	-0.3659	-0.2291	-0.3766	-0.0267	-0.0713	-0.147
LE	0.0159	-0.2887	-0.1471	-0.1846	-0.2304	-0.0943	0.0311

SL	-0.0715	-0.0708	0.2317	-0.0799	0.0301	0.4995	0.1361
RSL	-0.2485	-0.0177	-0.1293	0.0641	-0.1539	-0.4356	-0.2212
MPE	0.0439	-0.2363	-0.2176	-0.1138	-0.2807	0.0587	-0.3258
ATS	0.1948	0.2625	-0.0129	0.2735	-0.028	-0.1283	-0.1085
size ac	0.1304	0.9617	0.5121	0.8594	0.3484	-0.1728	0.2531
size rv	0.0297	0.9672	0.5024	0.892	0.2867	-0.2307	-0.0039
size tc	0.1152	0.9916	0.5232	0.8773	0.3749	-0.1806	0.0826
size pe	0.1058	0.9839	0.5232	0.8618	0.3854	-0.2096	0.0561
size le	0.0935	0.9845	0.5203	0.8851	0.3417	-0.1971	0.0783
size sl	0.122	0.9705	0.5435	0.8437	0.3987	-0.1246	0.1571
size rsl	0.0913	0.991	0.4985	0.8871	0.3508	-0.2062	0.0722
size mpe	0.0817	0.9612	0.4868	0.8904	0.2738	-0.1744	-0.0165
size ats	0.1116	0.9874	0.4824	0.8971	0.318	-0.1919	0.0529
size/sched	0.101	0.8938	0.3733	0.7239	0.4679	-0.4396	-0.0271
ss^.75	0.1002	0.8864	0.3858	0.7231	0.4548	-0.4994	0.0037
size^.75	0.1113	0.99	0.5092	0.8753	0.3724	-0.1855	0.0944

Variable	RV	TC	PE	LE	SL	RSL	MPE
RV	1						
TC	0.1303	1					
PE	-0.0493	0.2606	1				
LE	0.4523	0.3713	0.5203	1			
SL	-0.2264	0.1186	-0.1298	-0.1687	1		
RSL	0.5346	0.4904	0.1609	0.4214	-0.105	1	
MPE	0.3232	0.027	0.4815	0.7202	-0.1551	0.122	1
ATS	0.2124	0.0005	0.027	0.1151	-0.2135	0.0286	0.2238
size ac	-0.0437	-0.1082	-0.4596	-0.3322	-0.0272	-0.0548	-0.3335
size rv	0.1972	-0.0071	-0.3374	-0.1874	-0.1505	0.0934	-0.1503
size tc	0.0154	-0.0298	-0.3894	-0.2863	-0.0757	0.0047	-0.2526
size pe	0.0349	-0.034	-0.3221	-0.2646	-0.0983	0.0191	-0.2246
size le	0.0617	-0.0275	-0.372	-0.2244	-0.1021	0.032	-0.1981
size sl	-0.0574	-0.0864	-0.4439	-0.3523	0.0668	-0.0601	-0.3113
size rsl	0.0379	-0.0431	-0.3889	-0.2788	-0.0905	0.0397	-0.2453
size mpe	0.1005	-0.0477	-0.341	-0.1846	-0.1345	0.0314	-0.0598
size ats	0.0497	-0.0357	-0.3643	-0.2572	-0.1166	0.0131	-0.1994
size/sched	0.1264	-0.0087	-0.2103	-0.2051	-0.1944	0.1499	-0.2545
ss^.75	0.1388	-0.0507	-0.2618	-0.2252	-0.209	0.1712	-0.2803
size^.75	0.0015	-0.077	-0.4136	-0.3126	-0.0733	-0.0186	-0.2634

Variable	ATS	size ac	size rv	size tc	size pe	size le	size sl
ATS	1						
size ac	0.1941	1					
size rv	0.3432	0.9249	1				
size tc	0.2542	0.9776	0.972	1			
size pe	0.2617	0.9656	0.9749	0.9946	1		
size le	0.2777	0.9711	0.9799	0.9962	0.994	1	
size sl	0.1647	0.9855	0.9256	0.9807	0.9687	0.9696	1
size rsl	0.2591	0.9747	0.9773	0.9982	0.9941	0.9963	0.9758
size mpe	0.321	0.9296	0.9735	0.9719	0.974	0.9829	0.9335

size ats	0.3739	0.9528	0.9826	0.9881	0.9847	0.9887	0.9516
size/sched	0.3507	0.831	0.9097	0.8872	0.9036	0.886	0.8303
ss^.75	0.3187	0.8561	0.9085	0.8951	0.9111	0.8964	0.8465
size^.75	0.2391	0.9829	0.9666	0.9983	0.9922	0.9938	0.9851

Variable	size rsl	size mpe	size ats	size/sched	ss^.75	size^.75
size rsl	1					
size mpe	0.974	1				
size ats	0.9896	0.9769	1			
size/sched	0.8955	0.848	0.9027	1		
ss^.75	0.9043	0.8578	0.9005	0.9886	1	
size^.75	0.9974	0.97	0.9848	0.881	0.8929	1

Appendix F. Comparison of Input Requirements for Popular Models vs. Available Inputs

Table F.1. Model Input Comparison for Record Number 23-1

(Shaded areas are where the records have reported parameters)

Variable	Sage	PRICE-S	SASET	REVIC	Softcost
Amount of Travel			XX	XX	
Application Complexity		XX		XX	XX
Automated Tool Support	XX	XX	XX	XX	XX
Average Staffing Level		XX			
Code Delivery Requirements					XX
Common SLOC		XX	XX	XX	
Concurrent H/W Development					XX
Contract Type				XX	
COTS S/W			XX		
Customer Experience					XX
Database Complexity					XX
Database size					XX
Development Methods Experience	XX	XX		XX	
Development Model				XX	
Development Standard					XX
Development System Experience	XX	XX		XX	XX
Development System Volitility	XX			XX	XX
Development team			XX		
Development Year				XX	
Developoment System Volitility		XX			
Display Requirements	XX	XX		XX	
Embedded Development System			XX		
H/W constraints			XX		
H/W experience			XX		
Host Virtual System	XX			XX	
Inherent Difficulty of Application	XX			XX	
Level of Complexity		XX	XX		
Lifecycle phase				XX	XX
Man Interaction			XX		
Memory Constraints	XX	XX	XX	XX	XX
Modern Practices Experience	XX	XX		XX	XX
Modularity of S/W			XX		
Months in Development				XX	
Mulitiple Site Development	XX	XX	XX	XX	XX
New SLOC		XX	XX	XX	
Number Customer Locations			XX		
Number of CSCI Interfaces			XX		
Number of S/W config items			XX		
Number of Shifts				XX	
Number of Workstation types			XX		

Operating Evironment		XX			
Organizational Interface Complexity					XX
Peak Staffing		XX			
Percent of Microcode			XX		
Personnel Capability	XX	XX		XX	XX
Personnel Experience	XX	XX		XX	XX
Personnel Resources			XX		
Productivity Factor		XX			
Programming Language		XX	XX	XX	
Quality Assurance Level	XX			XX	
Quality Assurance Level		XX			
Real Time	XX			XX	XX
Rehosting Requirements	XX	XX	XX	XX	XX
Requirements Volitility	XX	XX		XX	XX
Resource Dedication	XX			XX	XX
Resources/Support Location	XX			XX	XX
Reusability Requirements	XX	XX		XX	
Reuse Impact	XX				
Reused SLOC		XX	XX	XX	
S/W documentation			XX		
S/W interfaces			XX		
S/W language complexity			XX		
S/W requirements			XX		
Schedule		XX			
Scope of Support					XX
Security Level	XX	XX		XX	XX
Software interfaces			XX		
Specification Level	XX			XX	XX
System Architecture					XX
System Requirements			XX		
Target System Experience		XX		XX	
Target Virtual System Experience	XX			XX	
Team Programmin Experience		XX			
Team Programming Language Experience	XX		XX		XX
Technology Impacts			XX		
Terminal Responses	XX				
Test Level	XX	XX	XX	XX	
Timing Constraints	XX	XX	XX	XX	XX
Turnaround Time	XX			XX	
Use of Peer Reviews					XX
User Involvement					XX

Table F.2. Model Input Comparison for Record Number 23-2

(Shaded areas are where the records have reported parameters)

Variable	Sage	PRICE-S	SASET	REVIC	Softcost
Amount of Travel			XX	XX	
Application Complexity		XX		XX	XX
Automated Tool Support	XX	XX	XX	XX	XX
Average Staffing Level		XX			

Code Delivery Requirements					xx
Common SLOC		xx	xx	xx	
Concurrent H/W Development					xx
Contract Type				xx	
COTS S/W			xx		
Customer Experience					xx
Database Complexity					xx
Database size					xx
Development Methods Experience	xx	xx		xx	
Development Model				xx	
Development Standard					xx
Development System Experience	xx	xx		xx	xx
Development System Volitility	xx			xx	xx
Development team			xx		
Development Year				xx	
Developoment System Volitility		xx			
Display Requirements	xx	xx		xx	
Embedded Development System			xx		
H/W constraints			xx		
H/W experience			xx		
Host Virtual System	xx			xx	
Inherent Difficulty of Application	xx			xx	
Level of Complexity		xx	xx		
Lifecycle phase				xx	xx
Man Interaction			xx		
Memory Constraints	xx	xx	xx	xx	xx
Modern Practices Experience	xx	xx		xx	xx
Modularity of S/W			xx		
Months in Development				xx	
Mulitiple Site Development	xx	xx	xx	xx	xx
New SLOC		xx	xx	xx	
Number Customer Locations			xx		
Number of CSCI Interfaces			xx		
Number of S/W config items			xx		
Number of Shifts				xx	
Number of Workstation types			xx		
Operating Evironment		xx			
Organizational Interface Complexity					xx
Peak Staffing		xx			
Percent of Microcode			xx		
Personnel Capability	xx	xx		xx	xx
Personnel Experience	xx	xx		xx	xx
Personnel Resources			xx		
Productivity Factor		xx			
Programming Language		xx	xx	xx	
Quality Assurance Level	xx			xx	
Quality Assurance Level		xx			
Real Time	xx			xx	xx
Rehosting Requirements	xx	xx	xx	xx	xx

Requirements Volitility	XX	XX		XX	XX
Resource Dedication	XX			XX	XX
Resources/Support Location	XX			XX	XX
Reusability Requirements	XX	XX		XX	
Reuse Impact	XX				
Reused SLOC		XX	XX	XX	
S/W documentation			XX		
S/W interfaces			XX		
S/W language complexity			XX		
S/W requirements			XX		
Schedule		XX			
Scope of Support					XX
Security Level	XX	XX		XX	XX
Software interfaces			XX		
Specification Level	XX			XX	XX
System Architecture					XX
System Requirements			XX		
Target System Experience		XX		XX	
Target Virtual System Experience	XX			XX	
Team Programmin Experience		XX			
Team Programming Language Experience	XX		XX		XX
Technology Impacts			XX		
Terminal Responses	XX				
Test Level	XX	XX	XX	XX	
Timing Constraints	XX	XX	XX	XX	XX
Turnaround Time	XX			XX	
Use of Peer Reviews					XX
User Involvement					XX

Table F.3. Model Input Comparison for Record Number 42-1

(Shaded areas are where the records have reported parameters)

Variable	Sage	PRICE-S	SASET	REVIC	Softcost
Amount of Travel			XX	XX	
Application Complexity		XX		XX	XX
Automated Tool Support	XX	XX	XX	XX	XX
Average Staffing Level		XX			
Code Delivery Requirements					XX
Common SLOC		XX	XX	XX	
Concurrent H/W Development					XX
Contract Type				XX	
COTS S/W			XX		
Customer Experience					XX
Database Complexity					XX
Database size					XX
Development Methods Experience	XX	XX		XX	
Development Model				XX	
Development Standard					XX
Development System Experience	XX	XX		XX	XX
Development System Volitility	XX			XX	XX

Development team			xx		
Development Year				xx	
Developoment System Volitility		xx			
Display Requirements	xx	xx		xx	
Embedded Development System			xx		
H/W constraints			xx		
H/W experience			xx		
Host Virtual System	xx			xx	
Inherent Difficulty of Application	xx			xx	
Level of Complexity		xx	xx		
Lifecycle phase				xx	xx
Man Interaction			xx		
Memory Constraints	xx	xx	xx	xx	xx
Modern Practices Experience	xx	xx		xx	xx
Modularity of S/W			xx		
Months in Development				xx	
Mulitiple Site Development	xx	xx	xx	xx	xx
New SLOC		xx	xx	xx	
Number Customer Locations			xx		
Number of CSCI Interfaces			xx		
Number of S/W config items			xx		
Number of Shifts				xx	
Number of Workstation types			xx		
Operating Evironment		xx			
Organizational Interface Complexity					xx
Peak Staffing		xx			
Percent of Microcode			xx		
Personnel Capability	xx	xx		xx	xx
Personnel Experience	xx	xx		xx	xx
Personnel Resources			xx		
Productivity Factor		xx			
Programming Language		xx	xx	xx	
Quality Assurance Level	xx			xx	
Quality Assurance Level		xx			
Real Time	xx			xx	xx
Rehosting Requirements	xx	xx	xx	xx	xx
Requirements Volitility	xx	xx		xx	xx
Resource Dedication	xx			xx	xx
Resources/Support Location	xx			xx	xx
Reusability Requirements	xx	xx		xx	
Reuse Impact	xx				
Reused SLOC		xx	xx	xx	
S/W documentation			xx		
S/W interfaces			xx		
S/W language complexity			xx		
S/W requirements			xx		
Schedule		xx			
Scope of Support					xx
Security Level	xx	xx		xx	xx

Software interfaces			XX		
Specification Level	XX			XX	XX
System Architecture					XX
System Requirements			XX		
Target System Experience		XX		XX	
Target Virtual System Experience	XX			XX	
Team Programmin Experience		XX			
Team Programming Language Experience	XX		XX		XX
Technology Impacts			XX		
Terminal Responses	XX				
Test Level	XX	XX	XX	XX	
Timing Constraints	XX	XX	XX	XX	XX
Turnaround Time	XX			XX	
Use of Peer Reviews					XX
User Involvement					XX

Table F.4. Model Input Comparison for Record Number 44-1

(Shaded areas are where the records have reported parameters)

Variable	Sage	PRICE-S	SASET	REVIC	Softcost
Amount of Travel			XX	XX	
Application Complexity		XX		XX	XX
Automated Tool Support	XX	XX	XX	XX	XX
Average Staffing Level		XX			
Code Delivery Requirements					XX
Common SLOC		XX	XX	XX	
Concurrent H/W Development					XX
Contract Type				XX	
COTS S/W			XX		
Customer Experience					XX
Database Complexity					XX
Database size					XX
Development Methods Experience	XX	XX		XX	
Development Model				XX	
Development Standard					XX
Development System Experience	XX	XX		XX	XX
Development System Volitility	XX			XX	XX
Development team			XX		
Development Year				XX	
Developoment System Volitility		XX			
Display Requirements	XX	XX		XX	
Embedded Development System			XX		
H/W constraints			XX		
H/W experience			XX		
Host Virtual System	XX			XX	
Inherent Difficulty of Application	XX			XX	
Level of Complexity		XX	XX		
Lifecycle phase				XX	XX
Man Interaction			XX		
Memory Constraints	XX	XX	XX	XX	XX

Modern Practices Experience	xx	xx		xx	xx
Modularity of S/W			xx		
Months in Development				xx	
Mulitiple Site Development	xx	xx	xx	xx	xx
New SLOC		xx	xx	xx	
Number Customer Locations			xx		
Number of CSCI Interfaces			xx		
Number of S/W config items			xx		
Number of Shifts				xx	
Number of Workstation types			xx		
Operating Evironment		xx			
Organizational Interface Complexity					xx
Peak Staffing		xx			
Percent of Microcode			xx		
Personnel Capability	xx	xx		xx	xx
Personnel Experience	xx	xx		xx	xx
Personnel Resources			xx		
Productivity Factor		xx			
Programming Language		xx	xx	xx	
Quality Assurance Level	xx			xx	
Quality Assurance Level		xx			
Real Time	xx			xx	xx
Rehosting Requirements	xx	xx	xx	xx	xx
Requirements Volitility	xx	xx		xx	xx
Resource Dedication	xx			xx	xx
Resources/Support Location	xx			xx	xx
Reusability Requirements	xx	xx		xx	
Reuse Impact	xx				
Reused SLOC		xx	xx	xx	
S/W documentation			xx		
S/W interfaces			xx		
S/W language complexity			xx		
S/W requirements			xx		
Schedule		xx			
Scope of Support					xx
Security Level	xx	xx		xx	xx
Software interfaces			xx		
Specification Level	xx			xx	xx
System Architecture					xx
System Requirements			xx		
Target System Experience		xx		xx	
Target Virtual System Experience	xx			xx	
Team Programmin Experience		xx			
Team Programming Language Experience	xx		xx		xx
Technology Impacts			xx		
Terminal Responses	xx				
Test Level	xx	xx	xx	xx	
Timing Constraints	xx	xx	xx	xx	xx
Turnaround Time	xx			xx	

Use of Peer Reviews					XX
User Involvement					XX

Table F.5. Model Input Comparison for Record Number 50-1

(Shaded areas are where the records have reported parameters)

Variable	Sage	PRICE-S	SASET	REVIC	Softcost
Amount of Travel			XX	XX	
Application Complexity		XX		XX	XX
Automated Tool Support	XX	XX	XX	XX	XX
Average Staffing Level		XX			
Code Delivery Requirements					XX
Common SLOC		XX	XX	XX	
Concurrent H/W Development					XX
Contract Type				XX	
COTS S/W			XX		
Customer Experience					XX
Database Complexity					XX
Database size					XX
Development Methods Experience	XX	XX		XX	
Development Model				XX	
Development Standard					XX
Development System Experience	XX	XX		XX	XX
Development System Volitility	XX			XX	XX
Development team			XX		
Development Year				XX	
Developoment System Volitility		XX			
Display Requirements	XX	XX		XX	
Embedded Development System			XX		
H/W constraints			XX		
H/W experience			XX		
Host Virtual System	XX			XX	
Inherent Difficulty of Application	XX			XX	
Level of Complexity		XX	XX		
Lifecycle phase				XX	XX
Man Interaction			XX		
Memory Constraints	XX	XX	XX	XX	XX
Modern Practices Experience	XX	XX		XX	XX
Modularity of S/W			XX		
Months in Development				XX	
Mulitiple Site Development	XX	XX	XX	XX	XX
New SLOC		XX	XX	XX	
Number Customer Locations			XX		
Number of CSCI Interfaces			XX		
Number of S/W config items			XX		
Number of Shifts				XX	
Number of Workstation types			XX		
Operating Evironment		XX			
Organizational Interface Complexity					XX
Peak Staffing		XX			

Percent of Microcode			XX		
Personnel Capability	XX	XX		XX	XX
Personnel Experience	XX	XX		XX	XX
Personnel Resources			XX		
Productivity Factor		XX			
Programming Language		XX	XX	XX	
Quality Assurance Level	XX			XX	
Quality Assurance Level		XX			
Real Time	XX			XX	XX
Rehosting Requirements	XX	XX	XX	XX	XX
Requirements Volitility	XX	XX		XX	XX
Resource Dedication	XX			XX	XX
Resources/Support Location	XX			XX	XX
Reusability Requirements	XX	XX		XX	
Reuse Impact	XX				
Reused SLOC		XX	XX	XX	
S/W documentation			XX		
S/W interfaces			XX		
S/W language complexity			XX		
S/W requirements			XX		
Schedule		XX			
Scope of Support					XX
Security Level	XX	XX		XX	XX
Software interfaces			XX		
Specification Level	XX			XX	XX
System Architecture					XX
System Requirements			XX		
Target System Experience		XX		XX	
Target Virtual System Experience	XX			XX	
Team Programmin Experience		XX			
Team Programming Language Experience	XX		XX		XX
Technology Impacts			XX		
Terminal Responses	XX				
Test Level	XX	XX	XX	XX	
Timing Constraints	XX	XX	XX	XX	XX
Turnaround Time	XX			XX	
Use of Peer Reviews					XX
User Involvement					XX

Table F.6. Model Input Comparison for Records 50-2, 5, 8, 9, 10, 11, 12, 13, and 15

(Shaded areas are where the records have reported parameters)

Variable	Sage	PRICE-S	SASET	REVIC	Softcost
Amount of Travel			XX	XX	
Application Complexity		XX		XX	XX
Automated Tool Support	XX	XX	XX	XX	XX
Average Staffing Level		XX			
Code Delivery Requirements					XX
Common SLOC		XX	XX	XX	
Concurrent H/W Development					XX

Contract Type				xx	
COTS S/W			xx		
Customer Experience					xx
Database Complexity					xx
Database size					xx
Development Methods Experience	xx	xx		xx	
Development Model				xx	
Development Standard					xx
Development System Experience	xx	xx		xx	xx
Development System Volitility	xx			xx	xx
Development team			xx		
Development Year				xx	
Developoment System Volitility		xx			
Display Requirements	xx	xx		xx	
Embedded Development System			xx		
H/W constraints			xx		
H/W experience			xx		
Host Virtual System	xx			xx	
Inherent Difficulty of Application	xx			xx	
Level of Complexity		xx	xx		
Lifecycle phase				xx	xx
Man Interaction			xx		
Memory Constraints	xx	xx	xx	xx	xx
Modern Practices Experience	xx	xx		xx	xx
Modularity of S/W			xx		
Months in Development				xx	
Mulitiple Site Development	xx	xx	xx	xx	xx
New SLOC		xx	xx	xx	
Number Customer Locations			xx		
Number of CSCI Interfaces			xx		
Number of S/W config items			xx		
Number of Shifts				xx	
Number of Workstation types			xx		
Operating Evironment		xx			
Organizational Interface Complexity					xx
Peak Staffing		xx			
Percent of Microcode			xx		
Personnel Capability	xx	xx		xx	xx
Personnel Experience	xx	xx		xx	xx
Personnel Resources			xx		
Productivity Factor		xx			
Programming Language		xx	xx	xx	
Quality Assurance Level	xx			xx	
Quality Assurance Level		xx			
Real Time	xx			xx	xx
Rehosting Requirements	xx	xx	xx	xx	xx
Requirements Volitility	xx	xx		xx	xx
Resource Dedication	xx			xx	xx
Resources/Support Location	xx			xx	xx

Reusability Requirements	xx	xx		xx	
Reuse Impact	xx				
Reused SLOC		xx	xx	xx	
S/W documentation			xx		
S/W interfaces			xx		
S/W language complexity			xx		
S/W requirements			xx		
Schedule		xx			
Scope of Support					xx
Security Level	xx	xx		xx	xx
Software interfaces			xx		
Specification Level	xx			xx	xx
System Architecture					xx
System Requirements			xx		
Target System Experience		xx		xx	
Target Virtual System Experience	xx			xx	
Team Programmin Experience		xx			
Team Programming Language Experience	xx		xx		xx
Technology Impacts			xx		
Terminal Responses	xx				
Test Level	xx	xx	xx	xx	
Timing Constraints	xx	xx	xx	xx	xx
Turnaround Time	xx			xx	
Use of Peer Reviews					xx
User Involvement					xx

Bibliography

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

Cummings, Cheri, Michael Gallo, Pamela Johnson, Barbara Marsh-Jones, Jill von Kuegelgen, Software Development Estimating Handbook, The Naval Center for Cost Analysis, 1998.

DeMarco,Tom, Why Does Software Cost So Much?, Dorset House Publishing, New York NY, 1995.

Ferens, Daniel V. and David S. Christensen, "Software Cost Model Calibration and Validation--An Air Force Case Study," The Journal of Cost Analysis, Fall 1997, pp. 43-56.

Ferens, Daniel V. and David S. Christensen, "Calibrating Software Cost Models to Department of Defense Databases," February 1998.

Ferens, Daniel V., "Software Cost Estimation in the DoD Environment," American Programmer, 9:28-34, July 1996.

Galonsky, James C. 1995. Calibration of the PRICE-S Model. Unpublished masters thesis. Dayton OH: Air Force Institute of Technology.

Hu, Qing, Robert T. Plant, and David B. Hertz, Software Cost Estimation Using Economic Production Models, Journal of Management Information Systems, Volume 15, Number 1, Summer 1998, pp. 143-164.

Jones, T. Capers, Estimating Software Costs, McGraw-Hill Companies, Inc., New York NY, 1998.

Kressin, Robert K. 1995. Calibration of the Software Life Cycle Model (SLIM) to the Space and Missile Systems Center Software Database (SWDB). Unpublished masters thesis. Dayton OH, Air Force Institute of Technology.

Mannering, Virginia H., Gross Domestic Product: Fourth Quarter 2000, Bureau of Economic Analysis, U.S. Dept. of Commerce, January 31, 2000.

Marzo, David B. 1997. Calibration and Validation of the SAGE Cost/Schedule Estimating System to United States Air Force Databases. Unpublished masters thesis. Dayton OH, Air Force Institute of Technology.

Mertes, Karen R. 1996. Calibration of the CHECKPOINT Model to the Space and Missile Systems Center (SMC) Software Database (SWDB). Unpublished masters thesis. Dayton OH, Air Force Institute of Technology.

Rathmann, Kolin D. 1995. Calibration of the System Evaluation and Estimation of Resources Software Estimation Model (SEER-SEM) for the Space and Missile Systems Center. Unpublished masters thesis. Dayton OH, Air Force Institute of Technology.

Shrum, Thomas C. 1997. Calibration and Validation of the CHECKPOINT Model to the Air Force Electronic Systems Center Software Database. Unpublished masters thesis. Dayton OH, Air Force Institute of Technology.

Stukes, Sherry and Eric Nguyen, Space and Missile System Center Software Database, MCR Federal, Inc., 1999.

Southwell, Steven V. 1996. Calibration of the SoftCost-R Software Cost Model to the Space and Missile Systems Center (SMC) Software Database (SWDB). Unpublished masters thesis. Dayton OH, Air Force Institute of Technology.

Stutzke, Richard D., "Software Estimating Technology: A Survey", Crosstalk, May 1996.

Symons, Charles R., Software Sizing and Estimating, John Wiley & Sons, New York NY, 1991.

Vegas, Carl D. *1995.* Calibration of the Software Architecture Sizing and Estimation Tool (SASET). Unpublished masters thesis. Dayton OH, Air Force Institute of Technology.

Weber, Betty G. 1995. A Calibration of the REVIC Software Cost Estimating Model. Unpublished masters thesis. Dayton OH, Air Force Institute of Technology.

White, Edward D., Class Lecture, STAT 535, Statistics, School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB OH, Fall Quarter 1999.

Vita

Captain Anthony W. Fife graduated from Meridian High School in Meridian, Idaho in 1985. He honorably served an LDS mission in Orange County, California from 1987 to 1989. While attending Ricks College, he served as a legislative intern for the Idaho State Senate. He graduated in 1992 from Boise State University with a Bachelors of Business Administration (emphasis in Accounting). Then in 1999, he graduated from California State University after earning a Masters of Business Administration (emphasis in International Business).

Captain Fife entered the United States Air Force in 1994 and, upon graduating Officer Training School, was commissioned an officer. His first assignment was to Electronic Systems Center (ESC), where he worked in the Cheyenne Mountain Upgrade (CMU) System Program Office (SPO). He was responsible for budget execution and cost analysis for the Granite Sentry program. While at ESC, he also worked the 767 AWACS program as a cost analyst.

Captain Fife was then reassigned to Los Angeles Air Force Base (LAAFB) in 1997. While at LAAFB, he worked for the MILSATCOM Joint Program Office. While there, he led the cost estimating effort for the Advanced EHF (AEHF) communications satellite and presented the Air Force cost position for AEHF to the Office of the Secretary of Defense, Cost Analysis Improvement Group (OSD CAIG). He also initiated the estimating efforts for the Gapfiller satellite system.

Captain Fife then entered the Graduate Acquisition Management program at the Air Force Institute of Technology in September 1999. Upon graduation, Captain Fife will be assigned to the Air Force Cost Analysis Agency.

Captain Fife married in 1989 and has three wonderful children.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
20-03-2001	Master's Thesis	Aug 1999 – Mar 2001

1. TITLE AND SUBTITLE

PREDICTING SOFTWARE DEVELOPMENT COST
FOR COMMAND AND CONTROL SYSTEMS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

*** AUTHOR(S)**

Fife, Anthony, W., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAQ/ENV/01M-05

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In the past, the Department of Defense (DoD) has relied upon commercial software estimating tools. However, these tools are somewhat unreliable when it comes to estimating military systems, particularly Command and Control Systems.

The purpose of this study was to develop a parametric model using linear regression to estimate software development costs for DoD Command and Control systems. The developed model is unique in a few ways. First, the model is derived from Department of Defense command and control data. Second, while traditional models require volumes of variables to create estimates, the developed model only requires a few key variables to estimate the amount of effort necessary to complete a project. The key variables were selected through analyzing common variables used in software cost estimating.

15. SUBJECT TERMS

Software Estimating, Linear Regression, Software Model, Command and Control Systems

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
* REPORT	* ABSTRACT	c. THIS PAGE			Lt Col William K. Stockman, ENV
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	167	(937) 255-3636, ext 4798

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Form Approved
OMB No. 074-0188