INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

**AN INVESTIGATION INTO THE DETERMINANTS OF
SOFTWARE PRODUCT DEVELOPMENT TIME**

by:

Brian Moretton

**A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements of the degree of**

**Master of Management Studies
School of Business**

**Carleton University
Ottawa, Ontario
July 27, 1997
© copyright**

1997, Brian Moretton

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-27476-4

The undersigned recommend to the Faculty of Graduate Studies and
Research
acceptance of the thesis

submitted by Brian Moretton, B. Comm. (Honours).
in partial fulfillment of the requirements for the degree of Master of
Management Studies

—
Thesis Supervisor

—
Director, School of Business

Carleton University

July 27, 1997

AN INVESTIGATION INTO THE DETERMINANTS OF SOFTWARE PRODUCT DEVELOPMENT TIME

by:

Brian Moretton

Abstract

This research investigates the determinants of shorter software development time. For the purpose of this research, software development time is defined as the interval between the point in time at which personnel are first assigned to analyze software product requirements and the release of the product to customers. Ten hypotheses are examined using multivariate linear regression on data from North American companies with over 20 employees listed companies in the Industry Canada Strategis (<http://strategis.ic.gc.ca>), the CanadaIT (<http://www.canadaIT.com>), and the Tickit (<http://www.tickit.com>) databases.

In this study, the number of weeks spent defining product requirements prior to coding was found to reduce development time after this point. Sales and marketing involvement in the development process and the number of

system tests run were also found to shorten development time, while emphasis on unit testing lengthened it. In addition, the research also demonstrated that a company's prior experience developing a product has an important intervening effect on the determinants of software development time.

Acknowledgments

I would like to thank Carleton University and the School of Business for allowing an open and accessible environment in which to conduct this research. With this, I thank Ms. Jean Blair for her assistance during the past two years. I thank Professors A. J. Bailetti and Bill Jones for being on my supervisory committee and for providing invaluable comment and insights.

I owe everything I am and will be to my parents. Mom and Dad, you always taught me the value of hard work and I thank you for that. I hope you are as proud of me as I am of you. I love you guys.

The research process is long road of peaks and valleys, my fiancée, Tamra, was with me every step of the way. She was always there for me as a sounding board and to pick me up whenever I got down. Tam, you are and always will be my inspiration.

Last, but certainly not least, I thank my supervisor John Callahan. I thank John for treating me as a student and not a student with a disability. John made me work just as hard as any other student, I thank him for that. Every once in a while one meets someone who touches them and makes them a better person. John, thanks for making me a better person.

This thesis is dedicated to the memory of Samantha Vitone.

Table of Contents

Abstract.....	iii
Acknowledgments.....	v
Table of Contents	vi
List of Tables	viii
Chapter 1 - Introduction	1
1.1 Research Objective.....	3
1.2 Contributions to the Literature.....	4
1.3 Managerial Contributions	5
1.4 Organization of the Thesis	6
Chapter 2 - Research Questions	8
2.1 Planning	10
2.2 Supplier Involvement.....	11
2.3 Functional Involvement	13
2.4 Rewards	13
2.6 Testing	15
2.7 Project Leader.....	16
Chapter 3 - Research Design	19
3.1 Unit of Analysis	19
3.2 Sampling Frame.....	19
3.3 Sampling Method	19
3.4 Sample Size	20
3.5 Respondents	20
3.6 Questionnaire.....	21
3.7 Piloting the Questionnaire	21
3.8 Operationalizing the Research Variables.....	22
3.9 Data Analysis	26
Chapter 4 - Results	30
4.1 Response Rate	30
4.2 Independent Variables	31
4.3 Dependent Variable	35
4.4 Control Variables.....	35
4.5 Exploratory Data Analysis	36
Outliers in the Data	47
4.6 Results of Regression Analysis.....	48
Hypothesis 1	49
Hypothesis 2	51
Hypothesis 3	53
Hypothesis 4	66
Hypothesis 5	69
Hypothesis 6a	71
Hypothesis 6b	73
Hypothesis 6c.....	75

Hypothesis 7a	77
Hypothesis 7b	79
4.7 Results of Correlation Analysis	80
4.8 Summary of Significant Results from Regression and Correlation Analysis	83
4.9 Results from Stepwise Regression Analysis.....	85
4.10 Qualitative Analysis of: What project leaders want more control over and How they would improve their development processes.....	90
Chapter 5 - Discussion of Results.....	94
5.1 Insights for Researchers	94
5.2 Insights for Managers	105
5.3 Limitations of Study.....	107
References	109
Appendix A - Questionnaire	111
Appendix B - Mann-Whitney U Comparisons of Means.....	124
Appendix C - Definition of Variables	133

List of Tables

Table 1 - Range, Means, Modes, Standard Deviations and Number of Observations for Independent Variables	32
Table 2 - Range, Means, Modes, Standard Deviations and Number of Observations for Dependent Variables	35
Table 3 - Range, Means, Modes, Standard Deviations and Number of Observations for the Control Variables	35
Table 4 - Spearman Correlations for Total Cost and Total person-months ...	36
Table 5 - Spearman Correlation Matrix for Independent Variables for Full Sample.....	37
Table 6 - Spearman Correlation Matrix for Independent Variables for Full Sample.....	38
Table 7 - Spearman Correlation Matrix for Independent Variables for Full Sample.....	39
Table 8 - Spearman Correlation Matrix for Independent Variables for Low Experience Sample.....	40
Table 9 - Spearman Correlation Matrix for Independent Variables for Low Experience Sample.....	41

Table 10 - Spearman Correlation Matrix for Independent Variables for Low Experience Sample.....	42
Table 11 - Spearman Correlation Matrix for Independent Variables for High Experience Sample.....	43
Table 12 - Spearman Correlation Matrix for Independent Variables for High Experience Sample.....	44
Table 13 - Spearman Correlation Matrix for Independent Variables for High Experience Sample.....	45
Table 14 - Results of Regression Analysis for Hypothesis 1	49
Table 15 - Results of Regression Analysis for Alternative Hypothesis 1.....	50
Table 16 - Split-sample Analysis for Hypothesis 1	50
Table 17 - Results of Regression Analysis for Hypothesis 2	51
Table 18 - Split-sample Analysis for Hypothesis 2.....	52
Table 19 - Results of Regression Analysis for Hypothesis 3	53
Table 20 - Split-sample Analysis for Hypothesis 3.....	54
Table 21 - Functional Involvement in Defining Product Requirements	54
Table 22 - Functional Involvement in System Design	55
Table 23 - Functional Involvement in Coding.....	56
Table 24 - Functional Involvement in System Testing	56
Table 25 - Functional Involvement in Beta Testing	58

Table 26 - Sales and Marketing Involvement in the Development Process...	59
Table 27 - Split Sample Analysis Sales and Marketing Involvement in the Development Process.....	59
Table 28 - Low Experience and No Sales and Marketing Involvement in Development Process.....	61
Table 29 - Low Experience and Sales and Marketing Involvement in Development Process.....	62
Table 30 - Summary of Significant Differences Between Low Experience Cases with Sales and Marketing Involvement and Low Experience Cases without Sales and Marketing Involvement	63
Table 31 - High Experience and No Sales and Marketing Involvement in Development Process.....	64
Table 32 - High Experience and Sales and Marketing Involvement in Development Process.....	64
Table 33 - Summary of Significant Differences Between High Experience Cases with Sales and Marketing Involvement and High Experience Cases without Sales and Marketing Involvement	65
Table 34 - Results of Regression Analysis for Hypothesis 4	66
Table 35 - Results of Regression Analysis for Alternative Hypothesis 4.....	67
Table 36 - Split-sample Analysis for Hypothesis 4.....	68

Table 37 - Results of Regression Analysis for Hypothesis 5	69
Table 38 - Split-sample Analysis for Hypothesis 5	70
Table 39 - Results of Regression Analysis for Hypothesis 6a	71
Table 40 - Split-sample Analysis for Hypothesis 6a.....	72
Table 41 - Results of Regression Analysis for Hypothesis 6b	73
Table 42 - Split-sample Analysis for Hypothesis 6b.....	74
Table 43 - Results of Regression Analysis for Hypothesis 6c.....	75
Table 44 - Split-sample Analysis for Hypothesis 6c	76
Table 45 - Results of Regression Analysis for Hypothesis 7a	77
Table 46 - Split-sample Analysis for Hypothesis 7a.....	78
Table 47 - Results of Regression Analysis for Hypothesis 7b	79
Table 48 - Split-sample Analysis for Hypothesis 7b.....	80
Table 49 - Spearman Correlations between Independent Variables and Software Development Time (Outliers Included)	81
Table 50 - Spearman Correlations between Independent Variables and Software Development Time (7 Outliers Removed)	82
Table 51 - Summary of Significant Relationships between Independent Variables and Software Development Time.....	84
Table 52 - Results from Stepwise Regression Analysis for Full Sample.....	86

Table 53 - Results from Stepwise Regression Analysis for Low Experience

Sample.....	88
-------------	----

Table 54 - Results from Stepwise Regression Analysis for High Experience

Sample.....	89
-------------	----

Table 55 - What Project Leaders Want More Control Over and 93

Chapter 1 - Introduction

With competition becoming increasingly more intense, companies that can bring products to market faster will stand a greater chance of surviving into the next century. While much has been written on strategies to shorten development time, very few empirical studies have been conducted to validate these strategies. Two of the limited number of studies were conducted by Eisenhardt and Tabrizi (1995) and Zirger and Hartley (1996).

Eisenhardt and Tabrizi (1995) contrasted two strategies that enable firms to achieve fast adaptation through product innovation: the compression strategy and the experiential strategy. They found the compression strategy to be most appropriate when product development is a certain, predictable process and can be planned out as a series of steps. This process can then be accelerated by shortening the time of each step, overlapping successive steps, and rewarding developers for attaining the compressed schedule. They found the experiential strategy most appropriately applied when product development is highly uncertain. In this case the keys to fast development are being able to learn quickly about changing conditions and remaining

flexible in order to adapt to these changes. In effect the compression and experiential strategies are examining the role that a company's prior experience developing a product plays in the development process.

Zirger and Hartley (1996) categorized the determinants of new product development time into three areas: product strategy, development process, and development team structure variables. The authors believed that there were various determinants that affected each of these variables and, in turn, development time. Zirger and Hartley (1996) found that fast developers had teams that were cross-functional, dedicated to one project, included development time as a goal, and overlapped development activities more so than slow developers.

While the above research furthers understanding on how to reduce development time, its testing and applicability is limited to the automobile, computer hardware, and electronics industries in which the products are tangible and manufacturability is an important design consideration. Software can be reproduced at virtually no cost so that its manufacturability is not an issue. Software is not tangible. It is difficult to make the design of a software product visible either to designers or to customers. Software is malleable in the sense that it can be easily changed, and unlike tangible products it does not have to obey a limiting set of physical laws. As a result, software design

offers distinctive management challenges. Hence, the applicability of empirically derived results on the determinants of product development time to the development of software is not certain. This research is designed to address this issue. It utilizes and adapts many of the determinants of development time found in the literature to investigate their effects on software development time. This research also attempts to capture and further the understanding of the effect of prior experience a company has in developing like software products on these determinants.

1.1 Research Objective

Companies are being challenged to develop new, innovative products at an increased pace. Companies are constantly searching for factors that will speed up the development process. This research investigates the determinants of shorter software product development time. This research also attempts to capture and further understanding of the effects of prior experience a company has in developing like software products on these determinants. For the purpose of this research, software product development time is defined as the interval between the point in time at which

personnel are first assigned to analyze software product requirements and the release of the product to customers.

1. 2 Contributions to the Literature

This research offers a number of important contributions to the literature. First, the study provides further insight into the determinants of software product development time. Second, it determines how these determinants are affected by a company's prior experience developing a product.

Many researchers have faced difficulties in developing measures that accurately capture factors that are important to software development time because they have attempted to characterize the development process as a series of chronological stages. Often software engineers have had difficulties viewing their development process in this manner or have disagreed with the order of these stages. By identifying common activities that occur in software development instead of attempting to divide development process into stages, the measures used in this study more accurately reflect factors that are important in software development. In doing so, the questionnaire used can

be a model for future research into the determinants of software development time.

1.3 Managerial Contributions

This research identifies the determinants of software development time and provides insight into how their influence depends on the type of development project. This generates managerial insights because managers can gain an understanding of what determinants should be emphasized to decrease development time in software projects. Specifically, this study found that the number of weeks spent defining product requirements reduced development time after this point. Sales and marketing involvement in the development process and the number of system tests run were also discovered to shorten development time, while emphasis on unit testing caused it to increase. In addition, the research also demonstrated that a company's prior experience developing a product has an important intervening effect on the determinants of software development time. These are the types of concrete insights that the research intended to offer.

1.4 Organization of the Thesis

Chapter 2 provides the rationale for the hypotheses examined in this thesis. These hypotheses examine the affect that various determinants have on software development time.

Chapter 3 describes the research design employed in this thesis. This chapter describes the research method, the sample and data collection, and analysis plan.

Chapter 4 presents the results of the research. This chapter is divided into ten sections. Section 4.1 provides the response rate for the questionnaire. Sections 4.2 to 4.4 present the descriptive statistics. Sections 4.3 and 4.4 also present the results of the ordinary least squares regression analyses and correlation analyses among dependent and control variables. Section 4.5 provides an exploratory data analysis. Sections 4.6 and 4.7 present a summary of results of the ordinary least squares regression analyses and Spearman correlation analyses required to examine the hypotheses. Section 4.8 presents a summary of significant results from the ordinary least squares regression analyses and Spearman correlation analyses. Section 4.9 presents the results of a best-fit regression analysis. Section 4.10 provides a qualitative analysis of responses to questions that asked project leaders what aspects of development they would like more

control over and how they would improve their development process in the future. The results of the research and the limitations of the research approach are discussed in Chapter 5.

Chapter 2 - Research Questions

While much has been written on strategies to shorten development time, very few empirical studies have been conducted to validate these strategies. Eisenhardt and Tabrizi (1995) compare two strategies for the management of product development projects. They call these strategies compression and experiential. According to Eisenhardt and Tabrizi (1995), a compression strategy is most useful when product development is a certain, predictable process which can be planned out as a series of steps. The process can then be accelerated by shortening the time of each step, overlapping successive steps, and rewarding developers for attaining the compressed schedule. The authors associated the following determinants with the compression strategy: **planning, supplier involvement, Computer Aided Design (CAD) usage, project overlap, multifunctional teams, and rewards.**

Eisenhardt and Tabrizi (1995) argue that an experiential strategy is appropriate when developing a radically new product because, then, the development process is a highly uncertain path through foggy and shifting markets and technologies. In these circumstances, the key to fast development is rapidly building intuition and flexible options in order to learn

quickly about and to shift with uncertain environments. It is also imperative that the development process have structure and motivate the pace in these environments because uncertainty can create chaos. Eisenhardt and Tabrizi hypothesized that the following determinants gave structure to the new product development process and reduced development time: **multiple iterations, frequent milestones, testing and project leader power.**

Zirger and Hartley (1996) categorized the determinants of new product development time into three areas: product strategy, development process, and development team structure variables. Zirger and Hartley (1996) believed that a company's choice of product strategy that minimizes development time is affected by **incremental as opposed to radical product changes** and the **number of parts in a product.** With respect to the actual development process, Zirger and Hartley (1996) theorized that three determinants accelerate development time: **overlapping development activities, freezing product design early, and supplier involvement in the development process.** Finally, the authors tested five determinants that affect a development team's structure, which in turn affects development time: **functional involvement, the extent team members are dedicated to the project, co-location, decision-making autonomy, and time as a goal.** In

summary, Zirger and Hartley (1996) found that fast developers had teams that were cross-functional, dedicated to one project, included development time as a goal, and overlapped development activities more so than slow developers.

The thesis here is that several of the determinants identified in the new product development literature can be adapted and applied to software development to reduce the development time of software products. Software development time is defined as the interval between the point in time at which personnel are assigned to analyze product requirements and the release of the product to customers. This chapter presents these determinants and develops the hypotheses investigated in this thesis.

2.1 Planning

Often effective planning before actual new product development can reduce development time. By emphasizing planning, one can better understand and rationalize the development process (Gupta and Wilemon, 1990). This, in turn, allows for the elimination of many steps and sequencing of activities in an efficient order. Comprehensive planning can also reduce mistakes and shorten delays (Gupta and Wilemon, 1990). Development teams that spend more time in planning can enhance the appearance of being competent and well organized (Pfeffer, 1992). Planning can serve as a signal

of project quality (Feldman and March, 1981). For software development companies such as Microsoft, the planning phase of the product development process involves defining a product vision, specification, and schedule (Cusumano and Shelby, 1995). A vision statement developed during product planning identifies and prioritizes product features. Also during the planning phase, a specification document is created which defines the product's functionality, architectural issues, and component interdependencies. Based on this a schedule for developing the product is established. Extensive planning will also allow the product development team to gain a better understanding of product requirements early on in the development process. By spending time defining a product's requirements early in the development process, the development team will have a clear goal to work towards and will not waste time adding more functionality to the product later in the process. This leads to the hypothesis that:

Hypothesis 1 (H1): More time spent defining product requirements prior to the start of coding is associated with shorter software development time.

2.2 Supplier Involvement

Many authors have argued that another way to shorten development time is to delegate the execution of some of these steps to other organizations

through supplier involvement (Imai, Nonaka, and Takeuchi, 1985). Others believe that increasing the degree of part design conducted by suppliers decreases development time (Zirger and Hartley 1996). Supplier involvement during many stages of development reduces the workload of the focal team (Clark and Fujimoto, 1991). Supplier involvement also allows the focal team to focus on the execution of tasks that take advantage of their skills and competencies. Involving suppliers in many steps of the process also will allow them to catch problems such as contradictory specifications or unrealistic designs sooner in the process, when they are easier to fix. Mabert, Muth, and Schmenner (1992) found that the suppliers were involved in many stages of fast projects. Although these arguments seem valid, before determining the affect that supplier involvement has on development time, one must consider the nature of the supplier involvement. It is quite possible to have supplier involvement in the development process and not have them affect development time. In order to have a true affect on development time, suppliers must play an active role in the development process. This may entail being involved in making key decisions with regards to aspects such as product specifications, system design, resource allocation or development tools. This leads the hypothesis that:

Hypothesis 2 (H2): More supplier involvement in a development project is associated with shorter software development time.

2.3 Functional Involvement

Multifunctional teams are closely linked to successful product development since different development stages rely on different functional expertise (Clark and Fujimoto, 1991). These teams allow technical, marketing, and manufacturing activities to be integrated and permit downstream problems in functional areas to be recognized earlier in the process when they are easier to fix (Imai, Nonaka, and Takeuchi, 1985). Some theorize that increased functional involvement allows the development team to make better decisions in regards to the product's configuration because the strengths and weaknesses of various technical and marketing alternatives can be considered earlier in the development process (Zirger and Hartley 1996). Involving more functions early in a development project can also reduce wait time (Imai, Nonaka, and Takeuchi, 1985). This leads to the hypothesis that:

Hypothesis 3 (H3): Functional involvement in the development process is associated with shorter software development time.

2.4 Rewards

Rewards can have a particularly important effect on development time

(Eisenhardt and Tabrizi, 1995). Rewards can be tied to meeting a fixed schedule. This can focus the attention of the developers on the project at hand and on time-based performance rather than on other issues such as the technological sophistication of the product (Eisenhardt and Tabrizi, 1995).

This leads to the hypothesis that:

Hypothesis 4 (H4): Greater financial rewards for schedule attainment are associated with shorter software development time.

2.5 Load Builds

Multiple design iterations improve the chances of achieving a successful product and thus accelerates the process. This will also allow a quicker means of learning (Eisenhardt, 1989). Many iterations improve the cognitive ability to shift with new information (Eisenhardt, 1989).

A load build is a system level compilation of a software product which software engineers utilize to test the product and iron out bugs (Cusumano and Shelby, 1995). It is common to have many load builds during a software development project. The frequency of these builds depends on the particular needs of a project and the amount of time required to complete a build successfully. When developing a radically new product, more frequent load builds offers rapid feedback to the development team about how the product

is progressing. Each build allows developers to gain a better understanding of the product (Cusumano and Shelby, 1995). Microsoft intends each load build to be very stable, meaning that they could theoretically ship the product to customers after each build. Using a load build as a proxy for design iteration, this leads to the hypotheses that:

Hypothesis 5 (H5): More load builds are associated with shorter software development time.

2.6 Testing

Testing is closely associated with both design iterations and frequent project milestones. Testing allows for the recognition of problems with each iteration. Problems are thus discovered earlier in the process when they are easier to rectify (Gupta and Wilemon, 1990). This is particularly important in uncertain environments. Through trial and error learning, the understanding of the product among development team members is improved. Testing also reduces development time because the concrete data that can be gathered through testing provides a basis of discussion on how the product can be improved. In software development there are primarily two procedures to testing software. Many software producers emphasize product testing for the most part at the end of the development process. Others such as Microsoft

believe in continuously testing the product while developing the product.

Three important methods of testing software products during development are unit tests, system tests, and beta tests. Unit tests are used by developers to discover and fix bugs while coding. System tests are internal, laboratory type tests of versions of the whole product. Beta tests occur when the product is tested by an end user/customer often on their site. This leads to the hypotheses that:

Hypothesis 6a (H6a): More emphasis on unit tests is associated with shorter software development time.

Hypothesis 6b (H6b): More tests of the whole product are associated with shorter software development time.

Hypothesis 6 (H6c): More beta sites are associated with shorter software development time.

2.7 Project Leader

Project leaders can have a critical effect on the speed and productivity of the development process (Clark and Fujimoto, 1991). These leaders assist to accelerate product development by keeping the process focused. Such a complex process can lose its focus if team members lose sight of overall goals. Project leaders must mesh together firm competencies and strategies with the needs of the market to create an effective product concept (Bowen et al., 1995). In order to effectively accomplish development goals, two

important aspects which the project leader must have control over are authority over product specifications and annual evaluations of development personnel. This leads to the hypotheses that:

Hypothesis 7a (H7a): Greater project leader control over product specifications is associated with shorter development time.

Hypothesis 7b (H7b): Greater project leader control over annual evaluations of development personnel is associated with shorter development time.

In summary this research tests ten hypotheses:

Hypothesis 1 (H1): More time spent defining product requirements prior to the start of coding is associated with shorter software development time.

Hypothesis 2 (H2): More supplier involvement in a development project is associated with shorter software development time.

Hypothesis 3 (H3): Functional involvement in the development process is associated with shorter software development time.

Hypothesis 4 (H4): Greater financial rewards for schedule attainment are associated with shorter software development time.

Hypothesis 5 (H5): More load builds are associated with shorter software development time.

Hypothesis 6a (H6a): More emphasis on unit tests is associated with shorter software development time.

Hypothesis 6b (H6b): More tests of the whole product are associated with shorter software development time.

Hypothesis 6 (H6c): More beta sites are associated with shorter software development time.

Hypothesis 7a (H7a): Greater project leader control over product specifications is associated with shorter development time.

Hypothesis 7b (H7b): Greater project leader control over annual evaluations of development personnel is associated with shorter development time.

Chapter 3 - Research Design

3.1 Unit of Analysis

The unit of analysis is a software development project.

3.2 Sampling Frame

The sampling frame for the research consisted of all software development projects undertaken by companies in North America with over 20 employees listed in the Industry Canada Strategis (<http://strategis.ic.gc.ca>), the CanadaIT <http://www.canadaIT.com>) and the Tickit (<http://www.tickit.com>) databases.

3.3 Sampling Method

E-mail messages were sent to the companies listed in the databases with 20 or more employees. The researcher identified himself as a graduate student at Carleton University's School of Business collecting data for a Master's thesis. The researcher asked that the message be directed to a software project leader. Questionnaires were sent to all companies who indicated an interest in participating in the study. All completed questionnaire were from companies with over 20 employees and were included in the data

set.

3.4 Sample Size

The sample size for this study was 37 software development projects.

3.5 Respondents

Respondents were project leaders in charge of software development projects. Initial company contact persons were asked to direct the questionnaire to a project leader. They were selected as respondents because they were knowledgeable about the project and are also responsible for project control. Although it was requested that the questionnaire be directed to a project leader and most often was completed by a project leader, the researcher acknowledges that occasionally companies delegate completion of the questionnaire to someone who was not directly involved in development. The effect of this on the research results is uncertain, but it is argued that it is minimal because most of the data requested could be found in standard development process documentation. To control for this in the future, a question may ask the respondent to rate the extent to which the data requested from the questionnaire was captured in the development process documentation.

3.6 Questionnaire

Data was collected using the questionnaire shown in Appendix A.

3.7 Piloting the Questionnaire

The questionnaire was piloted on five leaders of software development projects to ensure it worked as intended. Leaders were asked for specific feedback on how the questionnaire can be improved. These leaders offered an ideal pilot group because in addition to being very familiar with software development, they were also Master's students at Carleton University. This was beneficial because not only were they knowledgeable about the software development process, they understood the difficulties in developing clear, concise questions to capture data that is meaningful and capture the desired facets of the development process. The pilot group was helpful in testing the length of the questionnaire and in suggesting ways to make questions more understandable to "real life" project leaders.

3.8 Operationalizing the Research Variables

To capture some of the measures, many authors have divided development projects into the following stages: predevelopment, conceptual design, product design, testing, process development, and production start-up. This research attempted to move away from dividing the development process into stages and identified five activities of software development which are common to all development processes. The development activities that this research used were: defining product requirements, system design, creation of source code, system testing, and beta testing. These activities were identified for their use in this research by combining and consolidating the main activities that occur in Boehm's waterfall and spiral software development methods (1976, 1988). For this research, the five activities are defined as follows:

Defining product requirements is the determination of the commercial (or customer) requirements for the product together with its functional specification.

System design is the definition of the product design at the level of the modules to be used and their interdependencies.

Creation of source code is exactly that - the creation of the source code for the software product in whatever languages that are used.

System testing is the internal, laboratory type testing of version's of the whole product.

Beta testing is the testing of pre-release versions of the product by an end user/customer often on their site.

The following describes the measures used in this study:

Product requirements was measured as the proportion of total development time spent defining product requirements. It was computed by dividing the total reported time (weeks) defining product requirements prior to the start of coding by total development time.

Supplier involvement was calculated as the number of product development activities in which suppliers participated. Involvement was defined as having one or more employees of a supplier as recognized members of the product development team, actively participating in team meetings. The respondent also rated on a 7-point scale how important the supplier's personnel were in making key decisions on aspects such as resource allocation, product specifications, and system design product during each activity (1 = Not Important, 7 = Very Important).

Functional involvement was measured by asking whether personnel from specific functions were involved on the development team during the five activities of product development identified above. Involvement was defined as having one or more functional employees as recognized members of the product development team, actively participating in team meetings. The functions were marketing/sales, product management, software design, customer services, software quality assurance, and software maintenance. The total number of functions involved in each activity was calculated. The final measure was calculated by adding the total number of functions involved in each activity.

Reward for schedule attainment was restricted to monetary rewards. It was measured by calculating what proportion schedule attainment bonuses formed of the total base pay for the project as a whole.

Number of load builds was measured directly by asking how many load builds occurred in the development of the product.

Test was measured three ways. First, the respondent was asked to indicate how many times the system was tested as a whole product and how many beta test sites there were for the product. A second measure was developed by asking the respondent to rate the emphasis place on unit testing

to the overall testing of the product. For this thesis unit tests, system tests, and beta tests were defined as follows:

Unit tests are used by developers to discover and fix bugs while coding

System tests are internal, laboratory type tests of versions of the whole product.

Beta test sites occur when the product is tested by an end user/customer often on their site.

Project leader control was measured in the questionnaire by asking respondents Likert scale questions on the extent of control they had over product specifications and annual evaluations of development personnel.

Experience was measured by asking the respondent to rate on a 7-point Likert scale the company's prior experience developing such a product. The anchors were 1-none and 7-considerable.

Development time is the dependent variable in this study. Software development time was defined as the interval between the point in time at which personnel are first assigned to analyze software product requirements and the release of the product to customers.

Project size is a control variable in this study. It was measured by both the total allocated head count in person-months at project start and the total cost of a project.

3.9 Data Analysis

Quantitative data: The quantitative data were analyzed using the following steps.

Step 1. Calculations were performed on the raw data to develop the derived measures for the independent, dependent and control variables used in this study.

Step 2. Descriptive summary statistics were calculated for all of the variables, including the means, medians, standard deviations and Spearman correlations among the variables.

Step 3. Exploratory data analysis was performed using Spearman correlations. Spearman correlations were used to examine the associations between the variables because the distributions of many of the variables, both dependent and independent, were not Normally

distributed. Also Spearman correlations are not greatly affected by data outliers.

Step 4. For each hypothesis the independent variables were individually regressed against software development time using project size as the control variable. Spearman correlations were used to examine the associations between software development time and the independent variables. Outliers and violations of assumption for each model were checked using standardized residuals. One tailed t tests were used since the hypotheses are examining directional associations.

Step 5. To test how a company's prior experience developing a product affected the hypotheses in this study, it was decided to perform a split sample analysis. The sample was split into two categories using the **experience** variable. The projects which were below the **experience** variable mean (4.4) were placed in the low experience category and those that were above this mean were placed in the high experience category. There were 17 low experience and 20 high experience

projects. When the seven outlier projects were removed, there were 15 low experience projects and 15 high experience projects. To test for differences between the two samples the Mann-Whitney U test was used. Unlike a t-test, this non-parametric test does not require variables to follow a normal distribution. The results of this test are in appendix B. The only significant difference between the low and high experience group was found for functional involvement in defining product requirements and the number of load builds. The high experience group scored higher means on both variables.

Step 5. Stepwise regression was performed to develop the optimum predictor models for software development time. The independent variables were regressed against software development time using project size as the control variable. One tailed t tests were used since the hypotheses are examining directional associations. Outliers and violations of assumption for each model were checked using standardized residuals.

Qualitative Data: The responses to the final two open-ended questions in the questionnaire were analyzed using these steps.

Step 1. The qualitative data from each respondent were reviewed to find similarities in the responses in order to form categories to classify the data.

Step 2. Each of the responses were classified into the categories which they best fit under. Categories were not mutually exclusive. Responses could be placed in more than one category.

Step 3. The number of responses under each category were tallied and presented in frequency tables.

Chapter 4 - Results

This chapter is divided into ten sections. Section 4.1 provides the response rate for the questionnaire. Sections 4.2 to 4.4 present the descriptive statistics. Sections 4.3 and 4.4 also present the results of the ordinary least squares regression analyses and correlation analyses among dependent and control variables. Section 4.5 provides an exploratory data analysis. Sections 4.6 and 4.7 present the results of the ordinary least squares regression analyses and correlation analyses required to examine the hypotheses. Section 4.8 presents a summary of significant results from the ordinary least squares regression analyses and correlation analyses. Section 4.9 presents the results of a best-fit regression analysis. Section 4.10 provides a qualitative analysis of responses to questions that asked project leaders what they would like more control over and how they would improve their development process in the future.

4.1 Response Rate

The researcher sent email messages to the presidents of 119 software development companies requesting respondents for a Master's thesis. Fifty-five project leaders of these companies agreed to complete the questionnaire.

Questionnaire were sent out and returned via email. In total, thirty-seven usable questionnaires were returned. Thus the response rate for this study was 31 percent. All company respondents followed the initial selection criteria of having over 20 employees. The industries served by these companies ranged from aerospace engineering to personal computer applications software. The two most frequent reasons for not responding were time constraints and unwillingness to divulge what was believed to be confidential development information.

4.2 Independent Variables

The following table (Table 1) presents the range, means, modes, standard deviations and number of observations for the independent variables used in the study. This table excludes the control variables used.

Table 1- Range, Means, Modes, Standard Deviations and Number of Observations for Independent Variables

Independent Variable	Range	Mean	Mode	S.D.	Number of Observations
Number of weeks spent defining product requirements	0 - 26	8.6	4.4	7.6	37
Proportion of development process spent defining product requirements	0 - .57	.203	0	.201	37
Supplier involvement in development process	0 - 28	8.5	0	9.6	37
Functional involvement					
• in all activities	4-30	11.7	8	5.6	37
• in defining product requirements	1-6	2.8	2	1.5	37
• in system design	0-6	2.0	2	1.4	37
• In coding	0-6	1.6	1	1.3	37
• in system testing	0-6	2.6	3	1.3	37
• in beta testing	0-6	2.5	1	1.6	37
Financial rewards for schedule attainment (% of base pay)	0 - 62.5	4	0	10.7	37
Number of load builds	0 - 600	84.1	3	120.5	37
Emphasis on unit testing	2 - 7	5.8	7	1.3	37
Number of system tests	2 - 120	22.1	4	31	37
Number of beta test sites	1 - 3000	110.7	4	495.6	37
Project leader control over product specifications	2 - 7	5.4	5	1.2	37
Project leader control over annual evaluations of development personnel	1 - 7	5.8	7	1.7	37
Prior experience developing a product	1 - 7	4.4	7	2.1	37

Further examination of the independent variables revealed that 30% of the projects had no third party supplier involvement.

Functional involvement throughout all the development process activities ranged from 4 to 30. Only one company had total functional involvement during all development process activities.

Functional involvement in defining product requirements ranged from 1 to 6. Eighty-three percent of the sample projects had two functions or more functions involved in defining product requirements. Eight percent of sample projects involved all functions in defining product requirements.

Functional involvement in system design ranged from 0 to 6. Sixty-eight percent of the sample projects had two functions or more functions involved in system design. Five percent of sample projects involved all functions in system design.

Functional involvement in coding ranged from 0 to 6. Forty-six percent of the sample projects had two functions or more functions involved in coding. Only 1 of sample projects involved all functions in coding.

Functional involvement in system testing ranged from 0 to 6. Seventy-four percent of the sample projects had two functions or more functions involved in system testing. Only 2 of sample projects involved all functions in

system testing.

Functional involvement in beta testing ranged from 0 to 6. Sixty-eight percent of the sample projects had two functions or more functions involved in beta testing. Only 2 of sample projects involved all functions in beta testing.

Financial rewards as a percentage of base pay ranged from 0% to 65%. 65 percent of the companies in the sample did not offer financial rewards for schedule attainment.

The number of load builds carried out ranged from 0 to 600. No load builds were carried out in one project in the sample. Seventy-three percent of the projects carried out ten or more load builds.

The number of system tests carried out ranged from 2 to 120. System tests were carried out in all projects in the sample. Forty-six percent of the projects carried out ten or more system tests.

The number of beta test sites carried out ranged from 1 to 3000. Beta test sites were used in all projects in the sample. Six of the projects had only one beta site. Forty-three percent of the projects used five or more beta test sites.

4.3 Dependent Variable

Table 2 presents the range, means, modes, standard deviations and number of observations for the dependent variable in this study, software development time.

Table 2 - Range, Means, Modes, Standard Deviations and Number of Observations for Dependent Variables

Dependent Variable	Range	Mean	Mode	S.D.	Number of Observations
Software development time	11.1 - 175.7	47.5	13	30.1	37

4.4 Control Variables

Table 3 presents the range, means, modes, standard deviations and number of observations for the two variables used to measure project size. The total cost of each project in the sample ranged from \$15,000 to \$5,000,000. The number of person months initially allocated to the projects ranged from 3 to 4,500.

Table 3 - Range, Means, Modes, Standard Deviations and Number of Observations for the Control Variables

Control Variable	Range	Mean	Mode	S.D.	Number of Observations
Total cost of project	15,000 - 5,000,000	708,216	500,000	908,211	37
Total person-months dedicated to the project	3 - 4,500	235	60	777	37

Table 4 shows the correlations among the two control variables. Calculations were performed with 37 observations for total cost and total person-months. The results in Table 4 show that total person-months is strongly associated with total cost. For this reason one of the control variables, total person months was excluded from the regression analyses.

Table 4 - Spearman Correlations for Total Cost and Total person-months

Control Variable	Total cost of project	Total person-months dedicated to the project
Total cost of project	1	
Total person-months dedicated to the project	.53***	1

p<.05; *p<.01

4.5 Exploratory Data Analysis

The correlation matrix for the independent variables is shown on the following nine tables (**p<.05; ***p<.01):

Table 5 - Spearman Correlation Matrix for Independent Variables for Full Sample

Independent variable	Weeks spent defining product requirements	Supplier involvement in development process	Functional involvement in all activities	Functional involvement in defining product requirements	Functional involvement in system design	Functional Involvement in coding
Supplier involvement in development process	-.07	1				
Functional involvement in all activities	.14	.20	1			
Functional involvement in defining product requirements	-.09	.13	.67***	1		
Functional involvement in system design	.06	.22	.80***	.46***	1	
Functional involvement in coding	.13	.10	.71***	.35***	.59***	1
Functional involvement system testing	.04	-.06	.68***	.32**	.46***	.60***
Functional involvement in beta testing	.14	.18	.66***	.41***	.40***	.22
Financial rewards for schedule attainment (% of base pay)	.18	.18	.11	.09	.09	.08
Number of load builds	-.14	.06	.43***	.27	.25	.31**
Emphasis on unit testing	.26**	-.28**	-.12	-.04	-.05	.05
Number of system tests	-.46***	-.01	.23	.13	.09	.17
Number of beta sites	-.09	.11	-.07	-.23	-.01	-.006
Project leader control over product specifications	.10	-.19	-.01	.06	.09	-.03
Project leader control over annual evaluations of development personnel	-.03	.12	.03	.31**	.08	-.14
Prior experience developing a product	-.26**	-.10	.25	.50***	-.01	-.06
Project Cost	.38***	.17	.17	.05	.08	.19

Table 6 - Spearman Correlation Matrix for Independent Variables for Full Sample

Independent variable	Functional involvement system testing	Functional involvement in beta testing	Financial rewards for schedule attainment (% of base pay)	Number of load builds	Emphasis on unit testing	Number of system tests
Functional involvement in beta testing	.25	1				
Financial rewards for schedule attainment (% of base pay)	.03	.23	1			
Number of load builds	.20	.46***	.06	1		
Emphasis on unit testing	-.07	-.10	-.05	-.20	1	
Number of system tests	.19	.32**	-.02	.26	-.32**	1
Number of beta sites	.25	-.24	-.30**	-.11	.003	.03
Project leader control over product specifications	.09	-.02	.13	-.20	.19	.10
Project leader control over annual evaluations of development personnel	.02	-.05	.03	-.18	.04	.09
Prior experience developing a product	.30**	.10	-.08	.38***	-.04	.12
Project Cost	.07	.04	.26	.13	.10	-.35

Table 7 - Spearman Correlation Matrix for Independent Variables for Full Sample

Independent variable	Number of beta sites	Project leader control over product specifications	Project leader control over annual evaluations of development personnel	Prior experience developing a product
Project leader control over product specifications	.03	1		
Project leader control over annual evaluations of development personnel	-.07	.41***		
Prior experience developing a product	.1	-.02	.13	
Project Cost	-.04	-.41	-.22	.15

Table 8 - Spearman Correlation Matrix for Independent Variables for Low Experience Sample

Independent variable	Weeks spent defining product requirements	Supplier involvement in development process	Functional involvement in all activities	Functional involvement in defining product requirements	Functional involvement in system design	Functional involvement in coding
Supplier involvement in development process	-.22	1				
Functional involvement in all activities	.15	.39	1			
Functional involvement in defining product requirements	-.04	.08	.70***	1		
Functional involvement in system design	.19	.36	.88***	.46**	1	
Functional involvement in coding	.45**	.30	.67***	.33	.54**	1
Functional involvement system testing	-.10	.36	.71***	.43**	.71***	.40**
Functional involvement in beta testing	.08	.22	.66***	.68***	.49**	.20
Financial rewards for schedule attainment (% of base pay)	-.03	.23	.66***	.68***	.49**	.20
Number of load builds	-.06	-.03	.28	.52**	.10	-.04
Emphasis on unit testing	.47**	-.50**	-.46**	-.43**	-.23	-.13
Number of system tests	-.74***	.32	.12	.34	-.02	-.12
Number of beta sites	-.12	.24	-.27	-.46**	-.01	-.11
Project leader control over product specifications	-.20	.33	.02	.19	-.18	-.07
Project leader control over annual evaluations of development personnel	-.40	.31	.29	.46**	.21	.12
Prior experience developing a product	-.18	-.05	-.11	.37	-.23	-.25
Project Cost	.50***	-.15	.03	-.08	.06	.10

Table 9 - Spearman Correlation Matrix for Independent Variables for Low Experience Sample

Independent variable	Functional involvement system testing	Functional involvement in beta testing	Financial rewards for schedule attainment (% of base pay)	Number of load builds	Emphasis on unit testing	Number of system tests
Functional involvement in beta testing	.24	1				
Financial rewards for schedule attainment (% of base pay)	-.001	.52**	1			
Number of load builds	-.15	.66***	.19	1		
Emphasis on unit testing	-.30	-.32	-.03	-.24	1	
Number of system tests	-.07	.37	.24	.40	-.51	1
Number of beta sites	.17	-.56***	-.43**	-.33	.23	-.17
Project leader control over product specifications	.14	.24	.32	.08	-.05	.08
Project leader control over annual evaluations of development personnel	.47**	.03	.06	-.11	-.17	.37
Prior experience developing a product	.10	-.01	-.08	.21	-.06	.12
Project Cost	-.07	-.05	.007	-.11	.36	-.60

Table 10 - Spearman Correlation Matrix for Independent Variables for Low Experience Sample

Independent variable	Number of beta sites	Project leader control over product specifications	Project leader control over annual evaluations of development personnel	Prior experience developing a product
Project leader control over product specifications	.18	1		
Project leader control over annual evaluations of development personnel	.20	.35		
Prior experience developing a product	-.11	.26	.09	
Project Cost	-.35	-.36	-.32	.14

Table 11 - Spearman Correlation Matrix for Independent Variables for High Experience Sample

Independent variable	Weeks spent defining product requirements	Supplier involvement in development process	Functional involvement in all activities	Functional involvement in defining product requirements	Functional involvement in system design	Functional Involvement in coding
Supplier involvement in development process	-.02	1				
Functional involvement in all activities	.29	-.02	1			
Functional involvement in defining product requirements	.06	.06	.68***	1		
Functional involvement in system design	.05	.10	.78***	.59**	1	
Functional involvement in coding	.02	-.06	.77***	.45**	.63	1
Functional involvement system testing	.03	-.50**	.53***	.22	.22	.72***
Functional involvement in beta testing	.27	.14	.64***	.27	.41**	.22
Financial rewards for schedule attainment (% of base pay)	.34	.15	.07	.02	.18	.14
Number of load builds	-.05	.15	.43**	-.09	.44**	.52**
Emphasis on unit testing	.20	-.24	.18	.12	.09	.20
Number of system tests	-.29	-.06	.24	.05	.15	.31
Number of beta sites	-.10	-.05	.02	-.16	-.07	.03
Project leader control over product specifications	.11	-.47**	.01	-.16	.25	.004
Project leader control over annual evaluations of development personnel	.21	.004	-.26	.12	-.02	-.42**
Prior experience developing a product	-.20	-.53***	.17	.23	.04	.18
Project Cost	.38**	-.30	.17	-.06	.11	.24

Table 12 - Spearman Correlation Matrix for Independent Variables for High Experience Sample

Independent variable	Functional involvement system testing	Functional involvement in beta testing	Financial rewards for schedule attainment (% of base pay)	Number of load builds	Emphasis on unit testing	Number of system tests
Functional involvement in beta testing	.14	1				
Financial rewards for schedule attainment (% of base pay)	-.02	.06	1			
Number of load builds	.24	.29	.09	1		
Emphasis on unit testing	.14	.02	-.09	-.24	1	
Number of system tests	.26	.32	-.20	.16	-.20	1
Number of beta sites	.23	-.08	-.19	.02	-.20	.18
Project leader control over product specifications	.03	-.12	.01	-.26	.34	.14
Project leader control over annual evaluations of development personnel	-.37**	-.15	.03	-.38**	.21	-.21
Prior experience developing a product	.42	-.01	-.11	.12	-.13	-.03
Project Cost	-.07	.06	.53***	.22	-.06	-.16

Table 13 - Spearman Correlation Matrix for Independent Variables for High Experience Sample

Independent variable	Number of beta sites	Project leader control over product specifications	Project leader control over annual evaluations of development personnel	Prior experience developing a product
Project leader control over product specifications	-.07	1		
Project leader control over annual evaluations of development personnel	-.34	.47**		
Prior experience developing a product	.21	.14	-.14	
Project Cost	.12	-.41	-.26	-.14

Although correlation does not indicate causation, a number of the above results are interesting and worth commenting on. All of the functional involvement variables are highly positively correlated with one another for all three samples. The number of load builds run is also highly correlated with functional involvement in coding and beta testing for the full and high experience samples. One way to get more involvement from areas besides software design, may be to run more load builds. Load builds are tangible versions of a software product which other functions can understand and test easily.

Another interesting result is that emphasis on unit testing is negatively correlated with the number of system tests run for the full and low experience

samples. Companies are either emphasizing unit testing or system testing, not both. Emphasis on unit testing and the number of beta sites are also negatively correlated with functional involvement in the development process for the low experience sample. It may be that when experience is limited, product testing is left to the software engineers in the form of unit testing and to the customer/end user in the form of beta testing. The latter is supported by the result that indicates that, for the low experience sample, functional involvement in beta testing is negatively correlated with the number of beta test sites. It appears that when a company has little experience developing a product the benefit of functional involvement in product testing may be limited.

System testing is negatively correlated with the number of weeks spent defining product requirements for all three samples. This may indicate that the more time a company spends defining product requirements the more confident they are about the functionality about the product and therefore, the less time spent system testing. It is also interesting to note that project cost and the number of system tests run are negatively correlated. Possibly, full system tests occur less often in large projects because of the time and difficulties involved in coordinating a large development team for a full system test.

Another interesting result is that project cost and project leader control over product specifications and annual evaluations of development personnel are negatively correlated for all samples. It appears that in larger projects project leaders have less control than in smaller ones.

Lastly, for the high experience sample, prior experience developing a product is negatively correlated with supplier involvement in the development process. This indicates that the more experience a company has developing a product, the less they rely on suppliers for development assistance.

Outliers in the Data

Since the sample size is not large enough to use conventional test for outliers, extreme points, graphical methods such as scatter plots and normal p-p graphs were used to find these data points. Seven outlier cases were discovered. These outliers were not a result of coding errors. Efforts were made to return to respondents in order to explain these points, but none of the respondents returned requests for explanation. The points in question were:

Number of beta sites - Case 31 = 3000

Financial rewards for schedule attainment - Case 7 - 62.5%

Number of load builds- Case 32 = 600

Project Cost- Case 37 = 5000000

Person-months- Case 23 = 4000, (37 = 1350, 22 = 1250 were not considered outliers)

Throughout the following analyses, any major discrepancies caused by the outliers will be reported.

4.6 Results of Regression Analysis

This section presents the results of the hypotheses that were tested in this thesis. The hypotheses were tested using ordinary least squares regression. Each of the independent variables against software development time with project cost being a control variable. The general equation to be examined is:

Software development time = $b_0 + b_1 * \text{Independent variable} + b_2 * \text{Project cost}$

Project cost will be the only control variable used since it is significantly correlated with number of person-months initially allocated to the project. All calculations were performed with 37 observations. One tailed t tests were used since the hypotheses are examining directional associations.

Hypothesis 1

Table 14 shows the regression results relevant to Hypothesis 1:

More time spent defining product requirements prior to the start of coding is associated with shorter software development time.

Table 14 - Results of Regression Analysis for Hypothesis 1

Software development time	
Adjusted R-square	.07
F (significance)	2.40 (.10)**
tb1 (significance) Proportion of weeks defining product requirements	-.58 (.55)
tb2 (significance) Project Cost	2.04 (.049)**

p<.05; *p<.01

The above table displays the obvious that larger projects in terms of cost take longer to complete. The table also indicates that there is no significant relationship between the number of weeks spent defining product requirements and development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

As an aside, the premise of whether or not more time spent defining product requirements prior to coding reduced development time after this point was tested. The results are shown in Table 15.

Table 15 - Results of Regression Analysis for Alternative Hypothesis 1

Rest of development process	Full Sample	Outliers Removed
Adjusted R-square	.17	.33
F (significance)	4.95 (.013)***	8.4 (.000)***
tb1 (significance) Proportion of weeks defining product requirements	-2.16 (.04)**	-2.64 (.013)**
tb2 (significance) Project Cost	2.29 (.0503)	3.07 (.004)***

p<.05; *p<.01

Once more the above table displays that larger projects in terms of cost take longer to complete. The table also indicates that there is a significant negative relationship between the number of weeks spent defining product requirements and the rest of the development process for the full model and when the seven outliers are removed.

The split sample analysis for hypothesis 1 is below.

Table 16 - Split-sample Analysis for Hypothesis 1

Software development time	Low experience	High experience
Adjusted R-square	.57	.02
F (significance)	11.79 (.000)***	1.22 (.32)
tb1 (significance) Number of weeks defining product requirements	-.10 (.93)	.165 (.87)
tb2 (significance) Project Cost	4.4 (.000)***	1.56 (.14)

p<.05; *p<.01

Table 16 shows no significant relationship between the number of weeks spent defining product requirements and software development time for either group. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Hypothesis 2

Table 17 shows the regression results relevant to Hypothesis 2:

More supplier involvement in a development project is associated with shorter software development time.

Table 17 - Results of Regression Analysis for Hypothesis 2

Software development time	
Adjusted R-square	.15
F (significance)	4.17 (.024)**
tb ₁ (significance) Supplier involvement in the development process	-1.86 (.071)
tb ₂ (significance) Project Cost	2.6 (.014)**

p<.05; *p<.01

Table 17 shows no significant relationship between supplier involvement in the development process and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

The split sample analysis for this hypothesis is below.

Table 18 - Split-sample Analysis for Hypothesis 2

Software development time	Low experience (full sample)	Low experience (no outliers)	High experience (full sample)
Adjusted R-square	.71	.48	.03
F (significance)	21 (.000)***	7.6 (.02)**	1.3 (.298)
tb1 (significance) Supplier involvement in the development process	-2.62 (.02)**	-2.06 (.062)	-.40 (.694)
tb2 (significance) Project Cost	6.00 (.000)***	3.7 (.000)***	1.61 (.126)

p<.05; *p<.01

Table 18 indicates a significant ($p<.05$) negative relationship between supplier involvement in the development process and software development time for low experience companies. With the seven outliers removed, a significant relationship still exists at $p<.1$. No significant relationship also exists for the high experience group. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Hypothesis 3

Table 19 shows the regression results relevant to Hypothesis 3:

Functional involvement in the development process is associated with shorter software development time.

Table 19 - Results of Regression Analysis for Hypothesis 3

Software development time	
Adjusted R-square	.07
F (significance)	1.33 (.112)
tb1 (significance) Functional involvement in the development process	.461 (.648)
tb2 (significance) Project Cost	2.16 (.039)**

p<.05; *p<.01

Table 19 shows no significant relationship between functional involvement in the development process and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

The split sample analysis for this hypothesis is below.

Table 20 - Split-sample Analysis for Hypothesis 3

Software development time	Low experience	High experience
Adjusted R-square	.58	.03
F (significance)	12.08 (.000)***	1.3 (.300)
tb1 (significance) Functional involvement in the development process	-.484 (.637)	.40 (.700)
tb2 (significance) Project Cost	4.91 (.000)***	1.47 (.159)

p<.05; *p<.01

Table 20 shows no significant relationship between functional involvement in the development process and software development time.

The following five tables will identify what effect functional involvement in each development activity has on software development time.

Table 21 - Functional Involvement in Defining Product Requirements

Software development time	
Adjusted R-square	.12
F (significance)	3.55 (.04)**
tb1 (significance) Functional involvement in defining product requirements	-1.54 (.133)
tb2 (significance) Project Cost	2.42 (.021)**

p<.05; *p<.01

Table 21 shows no significant relationship between functional involvement in defining product requirements and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Table 22 - Functional Involvement in System Design

Software development time	
Adjusted R-square	.06
F (significance)	2.21 (.125)
tb ₁ (significance) Functional involvement in the system design	.021 (.983)
tb ₂ (significance) Project Cost	2.09 (.044)**

p<.05; *p<.01

Table 22 shows no significant relationship between functional involvement in system design and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Table 23 - Functional Involvement in Coding

Software development time	
Adjusted R-square	.07
F (significance)	2.36 (.109)
tb1 (significance) Functional involvement in coding	.52 (.608)
tb2 (significance) Project Cost	1.98 (.559)

p<.05; *p<.01

Table 23 shows no significant relationship between functional involvement in coding and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Table 24 - Functional Involvement in System Testing

Software development time	
Adjusted R-square	.06
F (significance)	2.23 (.123)
tb1 (significance) Functional involvement in system testing	-.175 (.862)
tb2 (significance) Project Cost	2.11 (.042)**

p<.05; *p<.01

Table 24 shows no significant relationship between functional involvement in system design and software development time. The residuals

were correlated with software development time, indicating an explanatory variable is missing.

Table 25 - Functional Involvement in Beta Testing

Software development time	
Adjusted R-square	.07
F (significance)	2.35 (.11)
tb ₁ (significance) Functional involvement in beta testing	.501 (.619)
tb ₂ (significance) Project Cost	2.159 (.038)**

p<.05; *p<.01

Table 25 shows no significant relationship between functional involvement in system design and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

To discover what effect involvement from the individual functions had on development time regressions were run using only involvement in development activities from one function at a time. The significant results from these regressions are given below.

Table 26 - Sales and Marketing Involvement in the Development Process

Software development time	Full Sample	Outliers Removed
Adjusted R-square	.31	.35
F (significance)	8.9 (.000)***	8.9 (.000)***
tb1 (significance) Sales and Marketing involvement in defining product requirements	-3.44 (.001)***	-2.65 (.013)**
tb2 (significance) Project Cost	2.20 (.034)**	2.69 (.011)**

p<.05; *p<.01

Table 26 indicates a significant negative relationship ($p<.01$) between sales and marketing involvement in defining product requirements and software development time for the full sample and with the seven outliers removed. The split sample analysis is given below.

Table 27 - Split Sample Analysis Sales and Marketing Involvement in the Development Process

Software development time	Low experience	High experience
Adjusted R-square	.80	.02
F (significance)	33.4 (.000)***	1.22 (.3)
tb1 (significance) Sales and Marketing involvement in the defining product requirements	-4.01 (.001)***	-.16 (.872)
tb2 (significance) Project Cost	5.4 (.000)***	1.555 (.14)

p<.05; *p<.01

Table 27 indicates a significant negative relationship ($p < .01$) between sales and marketing involvement in defining product requirements and software development time for low experience companies. No significant relationship exists for high experience companies. The following tables split the sample into four sub-samples based on experience and sales and marketing involvement in the development process to examine how the means of certain variables differ within each of the sub-samples.

Table 28 - Low Experience and No Sales and Marketing Involvement in Development Process

Variable	Mean	S.D.	Minimum	Maximum	Number of Observations
Software Development time	94.8	48.03	46.6	175.7	5
Project Cost (\$)	730,000	591,000	50,000	1,600,000	5
Number of beta test sites	12	21.3	1	50	5
Financial rewards for schedule attainment (\$)	24,000	43,359	0	100,000	5
Number of load builds	5.6	5.13	1	12	5
Functional involvement in development process	7.8	1.79	6	10	5
Supplier involvement in development process	1.00	1.22	0	3	5
Number of system tests	3.6	.89	2	4	5

Table 29 - Low Experience and Sales and Marketing Involvement in Development Process

Variable	Mean	S.D.	Minimum	Maximum	Number of Observations
Software Development time	36.9	20.5	13	74	12
Project Cost (\$)	390,250	434,055	15,000	1,300,000	12
Number of beta test sites	42	124	1	435	12
Financial rewards for schedule attainment (\$)	7,041	14,270	0	45,000	12
Number of load builds	54.8	78.34	3	250	12
Functional involvement in development process	11.6	6.1	4	28	12
Supplier involvement in development process	9.83	8.2	0	27	12
Number of system tests	19.75	27.42	3	100	12

The Mann-Whitney U test was used to test for significant differences between the low experience sample who had sales and marketing involvement and the low experience group that did not have sales and marketing involvement. A summary of the significant results are given below.

Table 30 - Summary of Significant Differences Between Low Experience Cases with Sales and Marketing Involvement and Low Experience Cases without Sales and Marketing Involvement

Variable	
Software Development time	significant difference at $p < .05$
Project Cost (\$)	
Number of beta test sites	
Financial rewards for schedule attainment (\$)	
Number of load builds	significant difference at $p < .1$
Functional involvement in development process	
Supplier involvement in development process	significant difference at $p < .1$
Number of system tests	significant difference at $p < .05$

Table 31 - High Experience and No Sales and Marketing Involvement in Development Process

Variable	Mean	S.D.	Minimum	Maximum	Number of Observations
Software Development time	44.07	11.2	31.5	53	3
Project Cost (\$)	988,333	1,327,086	15,000	2,500,000	3
Number of beta test sites	22	17.21	3	35	3
Financial rewards for schedule attainment (\$)	83,333	144,337	0	250,000	3
Number of load builds	332	237	150	600	3
Functional involvement in development process	10.33	2.52	8	13	3
Supplier involvement in development process	14	14	0	28	3
Number of system tests	53	42.51	4	80	3

Table 32 - High Experience and Sales and Marketing Involvement in Development Process

Variable	Mean	S.D.	Minimum	Maximum	Number of Observations
Software Development time	41.75	18.29	11.1	87.1	17
Project Cost (\$)	876,823	1,160,747	20,000	5,000,000	17
Number of beta test sites	203.76	724.39	1	3000	17
Financial rewards for schedule attainment (\$)	35,147	120,469	0	500,000	17
Number of load builds	84.25	85.66	3	300	17
Functional involvement in development process	13.24	6.03	4	30	17
Supplier involvement in development process	9	9.5	0	28	17
Number of system tests	23.82	34.74	2	120	17

The Mann-Whitney U test was used to test for significant differences between the high experience sample who had sales and marketing involvement and the high experience group that did not have sales and marketing involvement. A summary of the significant results are given below.

Table 33 - Summary of Significant Differences Between High Experience Cases with Sales and Marketing Involvement and High Experience Cases without Sales and Marketing Involvement

Variable	
Software Development time	
Project Cost (\$)	
Number of beta test sites	
Financial rewards for schedule attainment (\$)	
Number of load builds	significant difference at $p < .05$
Functional involvement in development process	
Supplier involvement in development process	
Number of system tests	

The above tables demonstrate the importance of sales and marketing to defining product requirements. Interestingly, it is the low experience, no sales and marketing involvement projects which perform poorly on the

measures examined. The average software development time is more than two times higher for these projects when compared to that of the next lowest group. The Mann-Whitney U test demonstrates that the presence of sales and marketing in defining product requirements is far more important to low experience projects than to high experience projects. Low experience projects without sales and marketing involvement differ from low experience ones with sales and marketing involvement on four of the eight variables examined, while high experience projects without sales and marketing involvement differ from high experience ones with sales and marketing involvement on only one.

Hypothesis 4

Table 34 shows the regression results relevant to Hypothesis 4:

Greater financial rewards for schedule attainment are associated with shorter software development time.

Table 34 - Results of Regression Analysis for Hypothesis 4

Software development time	
Adjusted R-square	.09
F (significance)	2.7 (.081)
tb ₁ (significance) Financial rewards for schedule attainment	-.928 (.362)

tb2 (significance) Project Cost	2.18 (.036)**
---------------------------------	---------------

p<.05; *p<.01

Table 34 shows no significant relationship between financial rewards for schedule attainment and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Since most of the companies in the sample did not offer financial rewards, it was decided to test this hypothesis using only those that did.

Table 35 - Results of Regression Analysis for Alternative Hypothesis 4

Software development time	
Adjusted R-square	.04
F (significance)	.96 (.41)
tb1 (significance) Financial rewards for schedule attainment	-1.23 (.22)
tb2 (significance) Project Cost	.58 (.57)

p<.05; *p<.01

Table 35 shows no significant relationship between rewards and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

The split sample analysis for this hypothesis is below.

Table 36 - Split-sample Analysis for Hypothesis 4

Software development time	Low experience	High experience
Adjusted R-square	.59	.18
F (significance)	12.31 (.000)***	3.11 (.07)
tb1 (significance) Financial rewards for schedule attainment	-.639 (.533)	1.82 (.085)
tb2 (significance) Project Cost	4.68 (.000)***	.109 (.915)

p<.05; *p<.01

Table 36 shows no significant relationship between financial rewards for schedule attainment and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Hypothesis 5

Table 37 shows the regression results relevant to Hypothesis 5:

More load builds are associated with shorter software development time.

Table 37 - Results of Regression Analysis for Hypothesis 5

Software development time	
Adjusted R-square	.08
F (significance)	2.47 (.09)
tb ₁ (significance) Number of load builds	-.671 (.51)
tb ₂ (significance) Project Cost	2.2 (.034)**

***p<.05; **p<.01

Table 37 shows no significant relationship between the number of load builds run and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

The split sample analysis for this hypothesis is below.

Table 38 - Split-sample Analysis for Hypothesis 5

Software development time	Low experience	High experience
Adjusted R-square	.59	.05
F (significance)	13.00 (.000)***	1.49 (.254)
tb1 (significance) Number of load builds	-.941 (.363)	.7 (.493)
tb2 (significance) Project Cost	4.09 (.000)***	1.24 (.233)

p<.05; *p<.01

Table 38 shows no significant relationship between the number of load builds run and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Hypothesis 6a

Table 39 shows the regression results relevant to Hypothesis 6a:

More emphasis on unit tests is associated with shorter software development time.

Table 39 - Results of Regression Analysis for Hypothesis 6a

Software development time	Full Sample	No Outliers
Adjusted R-square	.17	.26
F (significance)	4.81 (.015)**	6.3 (.000)***
tb ₁ (significance) Emphasis on unit testing	2.14 (.04)**	1.71 (.098)
tb ₂ (significance) Project Cost	2.35 (.03)**	3.26 (.003)***

p<.05; *p<.01

Table 39 shows a significant positive relationship ($p<.05$) between emphasis on unit tests and software development time. With the seven outliers removed, Table 39 also shows a significant positive relationship ($p<.1$) between emphasis on unit tests and software development time.

The split sample analysis for this hypothesis is below.

Table 40 - Split-sample Analysis for Hypothesis 6a

Software development time	Low experience	High experience
Adjusted R-square	.63	.13
F (significance)	14.33 (.000)***	2.44 (.117)
tb1 (significance) Emphasis on unit testing	1.38 (.188)	1.47 (.148)
tb2 (significance) Project Cost	4.71 (.000)***	1.84 (.082)

p<.05; *p<.01

Table 40 shows no significant relationships between emphasis on unit testing and software development time for the split-sample. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Hypothesis 6b

Table 41 shows the regression results relevant to Hypothesis 6b:
More tests of the whole product are associated with shorter software development time.

Table 41 - Results of Regression Analysis for Hypothesis 6b

Software development time	Full Sample	No Outliers
Adjusted R-square	.20	.22
F (significance)	5.66 (.007)***	5.11 (.013)**
tb1 (significance) Number of system tests	-2.47 (.019)**	-1.06 (.297)
tb2 (significance) Project Cost	2.78 (.009)***	2.65 (.013)**

p<.05; *p<.01

Table 41 shows a significant negative relationship between system tests and software development time. When the seven outliers are removed, there is no significant relationship between system tests and software development time.

The split sample analysis for this hypothesis is below.

Table 42 - Split-sample Analysis for Hypothesis 6b

Software development time	Low experience	High experience
Adjusted R-square	.58	.12
F (significance)	12.27 (.000)***	2.25 (.136)
tb1 (significance) Number of system tests	-.609 (.552)	-1.348 (.195)
tb2 (significance) Project Cost	4.25 (.000)	1.99 (.063)

p<.05; *p<.01

Table 42 show no significant relationships between the number of system tests and software development time for the split-sample. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Hypothesis 6c

Table 43 shows the regression results relevant to Hypothesis 6c:

More beta sites are associated with shorter software development time.

Table 43 - Results of Regression Analysis for Hypothesis 6c

Software development time	
Adjusted R-square	.08
F (significance)	2.52 (.095)
tb ₁ (significance) Number of beta sites	-.738 (.466)
tb ₂ (significance) Project Cost	2.167 (.037)**

p<.05; *p<.01

Table 43 shows no significant relationship between the number of beta sites and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

The split sample analysis for this hypothesis is below.

Table 44 - Split-sample Analysis for Hypothesis 6c

Software development time	Low experience	High experience
Adjusted R-square	.58	.04
F (significance)	12.19 (.000)***	1.38 (.279)
tb1 (significance) Number of beta sites	.56 (.584)	-.545 (.593)
tb2 (significance) Project Cost	4.81 (.000)***	1.59 (.129)

p<.05; *p<.01

Table 44 shows no significant relationships between the number of beta sites and software development time for the split-sample. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Hypothesis 7a

Table 45 shows the regression results relevant to Hypothesis 7a:

Greater project leader control over product specifications is associated with shorter development time.

Table 45 - Results of Regression Analysis for Hypothesis 7a

Software development time	
Adjusted R-square	.08
F (significance)	2.58 (.09)
tb ₁ (significance) Project leader control over product specifications	-.808 (.425)
tb ₂ (significance) Project Cost	1.83 (.079)

p<.05; *p<.01

Table 45 shows no significant relationship between project leader control over product specifications and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

The split sample analysis for this hypothesis is below.

Table 46 - Split-sample Analysis for Hypothesis 7a

Software development time	Low experience	High experience
Adjusted R-square	.59	.03
F (significance)	12.79 (.000)***	1.32 (.292)
tb1 (significance) Project leader control over product specifications	-.869 (.40)	-.452 (.657)
tb2 (significance) Project Cost	4.4 (.000)***	1.41 (.176)

p<.05; *p<.01

Table 46 shows that there is no significant relationships between project leader control over product specifications and software development time for the split-sample. The residuals were correlated with software development time, indicating an explanatory variable is missing.

Hypothesis 7b

Table 47 shows the regression results relevant to Hypothesis 7b:

Greater project leader control over annual evaluations of development personnel is associated with shorter development time.

Table 47 - Results of Regression Analysis for Hypothesis 7b

Software development time	
Adjusted R-square	.10
F (significance)	2.99 (.0636)
tb1 (significance) Project leader control over annual evaluations of development personnel	-1.17 (.249)
tb2 (significance) Project Cost	1.98 (.056)

p<.05; *p<.01

Table 47 shows no significant relationship between project leader control over annual evaluations of development personnel and software development time. The residuals were correlated with software development time, indicating an explanatory variable is missing.

The split sample analysis for this hypothesis is below.

Table 48 - Split-sample Analysis for Hypothesis 7b

Software development time	Low experience	High experience
Adjusted R-square	.62	.02
F (significance)	14.00 (.000)***	1.21 (.323)
tb1 (significance) Project leader control over annual evaluations of development personnel	-1.25 (.226)	.014 (.989)
tb2 (significance) Project Cost	4.99 (.000)***	1.51 (.15)

p<.05; *p<.01

Table 48 shows no significant relationships between project leader control over annual evaluations of development personnel and software development time for the split-sample. The residuals were correlated with software development time, indicating an explanatory variable is missing.

4.7 Results of Correlation Analysis

Table 49 shows the Spearman correlations between the independent variables and software development time with the outliers included in the sample. Table 50 shows the Spearman correlations between the independent variables and software development time when the seven outliers were removed from the sample.

Table 49 - Spearman Correlations between Independent Variables and Software Development Time (Outliers Included)

Independent Variable	Software development time (full sample)	Software development time (low experience)	Software development time (high experience)
Number of weeks spent defining product requirements	.50***	.72***	.40**
Number of weeks spent defining product requirements/Project Cost	-.20	-.43**	-.06
Number of weeks spent defining product requirements/development time	-.12	-.30	-.06
Supplier involvement in development process	-.10	-.30	.10
Functional involvement			
• in all activities	.11	.008	.33
• in defining product requirements	-.12	-.07	-.05
• in system design	.14	.06	.20
• In coding	.32**	.28	.38**
• in system testing	.007	-.19	.18
• in beta testing	.07	-.09	.28
Sales and marketing involvement in defining product requirements	-.43***	-.66***	-.13
Financial rewards for schedule attainment (% of base pay)	.18	-.09	.54***
Number of load builds	.12	-.27	.52***
Emphasis on unit testing	.19	.41***	.03
Number of system tests	-.40***	-.62***	-.20
Number of beta test sites	-.29**	-.41***	-.12
Project leader control over product specifications	-.34**	-.43***	-.25
Project leader control over annual evaluations of development personnel	-.31**	-.35**	-.21
Project Cost	.62***	.83***	.58***

p<.05; *p<.01

Table 50 - Spearman Correlations between Independent Variables and Software Development Time (7 Outliers Removed)

Independent Variable	Software development time (full sample)	Software development time (low experience)	Software development time (high experience)
Number of weeks spent defining product requirements	.47***	.66***	-.35**
Number of weeks spent defining product requirements/Project Cost	-.21	-.27	-.24
Number of weeks spent defining product requirements/development time	-.08	-.03	-.008
Supplier involvement in development process	-.06	-.18	.10
Functional involvement <ul style="list-style-type: none"> • in all activities • in defining product requirements • in system design • In coding • in system testing • in beta testing 	.08 -.24 .08 .37*** .08 .04	.00 .03 .08 .43*** -.08 .05	.03 -.50*** -.05 .28** .20 .06
Sales and marketing involvement in defining product requirements	-.48***	-.59***	-.41**
Financial rewards for schedule attainment (% of base pay)	.29	.20	.37**
Number of load builds	.17	-.21	.63***
Emphasis on unit testing	.07	.20	-.04
Number of system tests	-.41***	-.54**	-.34
Number of beta test sites	-.29	-.55**	.07
Project leader control over product specifications	-.35**	-.30**	-.41**
Project leader control over annual evaluations of development personnel	-.25	-.17	.23
Project Cost	.61	.75***	.62

p<.05; *p<.01

4.8 Summary of Significant Results from Regression and Correlation Analysis

Table 51 provides a summary of significant results from the regression and Spearman correlation analyses.

Table 51 - Summary of Significant Relationships between Independent Variables and Software Development Time

Independent Variable	Regression Analysis	Correlation analysis
Number of weeks spent defining product requirements	Negative relationship at $p < .05$ with <u>rest</u> of development process	Positive relationship at $p < .01$ for full and low experience and at $p < .05$ for high experience samples
Number of weeks spent defining product requirements/Project Cost		Negative relationship at $p < .05$ for low experience sample (outliers not removed)
Supplier involvement in development process	Negative relationship at $p < .05$ for low experience sample and at $p < .1$ with outliers removed	
Functional involvement <ul style="list-style-type: none"> in defining product requirements in coding 	Negative relationship at $p < .01$ for only sales and marketing for full and low experience samples	Negative relationship at $p < .01$ for only sales and marketing for full and low experience samples Positive relationship at $p < .05$ for full, low, and high experience samples (not significant for low experience sample when outliers removed)
Financial rewards for schedule attainment (% of base pay)		Positive relationship at $p < .01$ for high experience sample (at $p < .05$ with outliers removed)
Number of load builds		Positive relationship at $p < .01$ for high experience sample
Emphasis on unit testing	Positive relationship at $p < .05$ for full sample (outliers not removed)	Positive relationship at $p < .01$ for high experience sample
Number of system tests	Negative relationship at $p < .05$ for full sample (outliers not removed)	Negative relationship at $p < .01$ for full and low experience samples
Number of beta test sites		Negative relationship at $p < .05$ for full and at $p < .01$ for low experience samples
Project leader control over product specifications		Negative relationship at $p < .05$ for full and at $p < .01$ low experience samples
Project leader control over annual evaluations of development personnel		Negative relationship at $p < .05$ for full and low experience samples (not significant when outliers removed)

4.9 Results from Stepwise Regression Analysis

Stepwise regression analysis was chosen to develop optimum predictor models for software development time. Stepwise regression was chosen because it minimizes the influence of missing variables and maximizes the predictive power of variables used in this study. All the independent and control variables were entered into the stepwise regression. Regressions for the full, low experience, and high experience samples were run. One-tailed t tests were used since the hypotheses are examining directional associations. The p-entry value was set at $p < .05$ and p-exit value was set at $p < .1$.

Table 52 - Results from Stepwise Regression Analysis for Full Sample

Software development time	Full Sample	Outliers Removed
Adjusted R-square	.20	.42
F (significance)	5.65 (.007)***	8.23 (.000)***
Prior experience developing a product	did not enter model	-2.76 (.01)***
Supplier involvement in development process	did not enter model	-2.36 (.03)**
Number of system tests	-2.47 (.019)**	did not enter model
Project Cost	2.78 (.008)***	4.46 (.000)***

p<.05; *p<.01

This optimum predictor model accounts for 20% of the variance. A negative association at $p<.05$ was found between software development time and the number of system tests run. Cost of project was also found significant. The residuals were correlated with software development time, indicating an explanatory variable is missing.

A stronger overall predictor model is developed when the seven outlier projects are removed from the sample. This optimum predictor model accounts for 42% of the variance. Prior experience developing a product was found to be negatively correlated at $p<.01$, while supplier involvement in the development process was found to be negatively correlated at $p<.05$. Cost of

project was also found significant. Number of system tests was not significant.

Table 53 - Results from Stepwise Regression Analysis for Low Experience Sample

Software development time	Outliers Included	Outliers Removed
Adjusted R-square	.71	.71
F (significance)	21.00 (.000)***	12.63 (.000)***
Prior experience developing a product	did not enter model	-3.25 (.000)***
Supplier involvement in development process	-2.62 (.02)**	-2.83 (.016)**
Project Cost	6.00 (.000)***	5.13 (.000)***

p<.05; *p<.01

The optimum predictor model for the low experience sample accounts for 71% of the variance. A negative association at $p<.05$ was found between software development time and supplier involvement in the development process. Cost of project was also found significant. The residuals were correlated with software development time, indicating an explanatory variable is missing.

With the seven outliers removed from the sample, the predictor model for the low experience sample once again accounts for 71% of the variance. Prior experience developing a product and supplier involvement in the development process were negatively correlated with software development time at $p<.01$ and $p<.05$, respectively. Cost of project was significant at $p<.01$.

Table 54 - Results from Stepwise Regression Analysis for High Experience Sample

Software development time	Outliers Included	Outliers Removed
Adjusted R-square	.23	.26
F (significance)	6.58 (.019)**	6.27 (.025)**
Project Cost	did not enter model	2.50 (.025)**
Financial rewards for schedule attainment	2.57 (.019)**	did not enter model

*p<.1 **p<.05; ***p<.01

The optimum predictor model for the high experience sample accounts for 23% of the variance. A positive association at $p<.05$ was found between software development time and financial rewards for schedule attainment. Cost of project was not found significant. The residuals were correlated with software development time, indicating an explanatory variable is missing.

When the seven outliers are removed, the predictor model for the high experience sample accounts for 26% of the variance. Only cost of project was significant at $p<.05$. The residuals were correlated with software development time, indicating an explanatory variable is missing. It is interesting to note that the optimum predictor models for software development time for the high and low experience samples are distinctly different. This indicates that prior experience developing a product has great

impact on and must be considered when examining the determinants of software development time.

4.10 Qualitative Analysis of: What project leaders want more control over and How they would improve their development processes

This section provides a qualitative analysis of responses to questions that asked project leaders what they would like more control over (Question #32) and how they would improve their development process in the future (Question #33). Analysis of these questions revealed that the responses could be classified under eleven categories. These categories were not mutually exclusive, meaning that some responses were classified under more than one category. The following lists the categories and provides a brief sample of the data which fell under each

Development Resources

1. More control over development budget
2. Better financial resource management
3. More control over who was assigned to project

Development Time Frames

1. Better estimates of development time needed
2. More control over development process schedule
3. Clear coding time frames and clear requirements contract
4. Creation of requirements determination prior to coding
5. Clear delivery time frames
6. Clear project priorities

Defining Product Requirements

1. More initial decisions regarding quality vs. features
2. Make decisions based on technical merit as opposed to perceived marketing requirements
3. More marketing and product management input into the requirements phase
4. More clearly defined end user requirements

Product Specifications

1. More control over specifications
2. More control over system level specification and initial design architecture
3. Agree on final specification issues sooner in project
4. A less dynamic product specification
5. Spend more time up front in the detailed design of the product

Marketing

1. Better initial marketing input
2. More marketing

Development Tools

1. Better hardware and operating system platforms
2. developed in C++
3. Limitations on various platforms and architectures severely limited alternatives for development, and dramatically increased costs
4. Faster Hardware
5. Wait until the new development tools used had matured before starting the project
6. Better knowledge of emerging new development tools from suppliers

Training of Development Personnel

1. More scheduled training
2. Better overall training

Development Site

1. All development at one site

Product Testing

1. More code reviews
2. A more limited first version of the product and an earlier release to more beta sites
3. More committed beta customers
4. More testing platforms
5. Higher levels of performance testing using the original development staff
6. Fewer builds of the product, allowing more full tests
7. Quicker response from the end user with data to test sooner
8. More involvement of test designers and testers earlier; make them part of the process early
9. Stronger participation by the Customer in testing and evaluation prior to installation

Supplier Involvement

1. More control over supplier performance
2. Better knowledge of supplier's products

End User/Customer Involvement

1. More involvement from customers throughout development process
2. More involvement from customers throughout requirements gathering
3. More involvement from customers throughout product testing
4. More control over what customer staff was assigned to the project
5. More control over what customer personnel was assigned to project

Table 55 offers a breakdown down of the number of responses in each category.

**Table 55 - What Project Leaders Want More Control Over and
How They Would Improve Their Development Processes**

Category	Question #32- What they want more control over	Question #33- How they would improve their development processes	Total
Development resources	5	2	7
Development time frames	7	5	12
Defining product requirements	5	6	11
Product specifications	7	6	12
Marketing	0	2	2
Development tools	1	5	6
Training of development personnel	0	2	2
Development site	1	0	1
Product testing	1	10	11
Supplier involvement	2	2	4
End user/Customer Involvement	3	7	10

Chapter 5 - Discussion of Results

The results of this thesis provide many interesting insights for both researchers and managers. Section 5.1 highlights these results in terms of their benefit to researchers, while section 5.2 provides insight into how the results will impact managers and improve the development process. Section 5.3 will then discuss the limitations of this study.

5.1 Insights for Researchers

Before highlighting the actual results of this study, the process by which the data was gathered and the results arrived upon will be discussed in an attempt to offer future researchers insights on improving their research process. This study began to utilize the extraordinary potential that the Internet offers researchers. The Internet offers researchers easy access to an ever increasing research population. This study utilized software company databases from three different websites to solicit respondents. Email allowed for easy communication with respondents. The questionnaire was prepared using a relatively new software program, Decisive Survey. Decisive allows for the creation, management, and analysis of email surveys. Questionnaires were sent to respondents and collected using this software package. Respondents commented on how easily having the questionnaire on email

allowed them to answer the questions quickly. Decisive also automatically recorded which respondents were sent and returned the questionnaire. This facilitated the sending out of reminders. Once the questionnaires were returned, Decisive not only performed elementary analysis on the data, but also allowed it to be exported into a more sophisticated statistical software package. This eliminated the time it would have taken to code the data and eliminated coding errors. The same holds for the qualitative data. The software easily allowed qualitative data to be electronically cut and pasted into a word processor.

In terms of the results themselves, many of the split-sample analyses suggest that a company's prior experience developing a product has a very important intervening effect on the determinants of software development time. Although it is agreed that splitting a sample diminishes the power of any results, one cannot disagree that these results are intriguing and offer many avenues for further investigation. In itself, the manner in which experience was measured offers a contribution to the literature. A careful examination of Eisenhardt and Tabrizi's (1995) compression and experiential strategies reveals that they were ex-post categorizations based on the type of product that a company developed. The authors classified mainframe hardware development as following a compression strategy because they believed that

the development process here was certain and predictable. PC hardware development was classified as following an experiential strategy because they felt that the development process here was uncertain and unpredictable. The argument that contradicts this is that if a company lacks prior experience developing a product, it does not matter what type of product is developed. This research allowed respondents to rate their own prior experience. This allowed a true examination of the intervening role of experience on the determinants of software development time.

While many of the results support findings in the literature, others contradicted them. The number of weeks spent defining product requirements prior to coding was negatively correlated with the time it took for the rest of the development process. This finding held true for companies of all experience levels. In some cases, especially in large companies, product requirements are handed down to the development team at project start, but in cases where the development team is responsible for defining product requirements, this finding supports Eisenhardt and Tabrizi's (1995) assertion that more time spent in up front planning prior to development will shorten development time. Interestingly many of the qualitative responses indicated that project leaders would like to have spent more time in and have more control over the defining of product requirements. They also indicated that they would like more end

user/customer input during this activity. Considering the results here, more research is needed to discover first, why more time is not spent defining product requirements now and second, how the end user/customer can become more active in the defining of product requirements and shorten development time.

In this study, supplier involvement in the development process was discovered to shorten development time for companies which have limited experience developing a product. If supplier involvement is defined in the traditional sense whereby suppliers assist developers by designing parts for the overall product, this result supports Zirger and Hartley's (1996) contention that increasing the degree of part design conducted by suppliers decreases development time. While in general, this also supports Imai et. al's (1985) theory that supplier involvement will expedite development time, it contradicts Eisenhardt and Tabrizi's (1995) hypothesis that supplier involvement will shorten development time for experienced companies. This suggests that the role of suppliers in software development differ from the role of suppliers in the auto industry, for example. Although some companies contract the creation of separate modules that comprise a software product, more often than not, the supplier relationship with software companies entails the provision of development tools. One project leader stated that, "with more

knowledge about what (their) suppliers are doing such as products they are working on and their scheduled release dates, (they) would have improved our product development.” Since these tools are constantly changing and improving, it is definitely important to maintain a close relationship with suppliers. This may be especially important for low experience companies because they may not be aware of the best development tools to use or may not have the resources to develop their own tools.

More research into the developer-supplier relationship in software development is necessary. Supplier involvement may mean different things to different respondents. More specific questions are required to understand supplier involvement in each of these circumstances. Respondents could be asked to outline whether suppliers directly affected the development of a software product in terms of developing modules for the product or whether they indirectly affected development by providing aspects such as development personnel, financial resources, or development tools. Examples of questions to elicit the nature and effect of supplier involvement might include:

- Did third party suppliers directly develop software modules for the product?

- Rate the quality of the modules developed by the third party supplier.
- Did third party suppliers provide personnel for the development team?
- Rate the contribution of third party supplier personnel to the project.
- Did third party suppliers provide financial resources for the development project?
- Did third party suppliers provide development tools for the development project?
- How much influence did third party suppliers have over the makeup of the development team?
- How much influence did third party suppliers have over product specifications?
- How much influence did third party suppliers have over defining product requirements?

The involvement of sales and marketing in defining product requirements was found to shorten development time, while overall functional involvement in the coding activity was positively correlated with software

development time. This partially upholds Zirger and Hartley's (1996) hypothesis that increasing the number of functions represented on the development team decreases development time. This finding more accurately supports Gupta and Wilemon (1990) and Clark and Wheelwright's (1992) assertion that early involvement of functional team members will shorten development time, while late involvement will invariably cause it increase. The logic here is that during the latter stages of development the product has already taken shape and therefore any new ideas or opinions that may be incorporated into the product at this point may increase development time. Often developers want to create the perfect product and lose sight of the overall task at hand. This results in wasted time continuously adding new features.

Interestingly, financial rewards for schedule attainment was positively correlated with software development time for high experience companies. This finding directly contrasts the Eisenhardt and Tabrizi (1995) hypothesis that rewards will speed development time. It should, however, be noted that the distribution of financial rewards for schedule attainment in this study is very limited and as a result, is of limited use. Very few companies offered financial rewards for schedule attainment and for those that did, with outliers

removed, financial rewards for schedule attainment accounted for only 2% of base pay.

Also, contradictory to the Eisenhardt (1989) assertion that more design iterations (load builds) reduce development time, load builds were positively correlated with software development time for high experience companies. This somewhat follows the Eisenhardt and Tabrizi (1995) logic that while frequent iterations may speed development time for low experience companies, these iterations in fact increase development time for high experience companies. The argument here is that while low experience companies may learn from frequent iterations, they have no impact on high experience ones. It may be that when companies have considerable experience developing a product they do not gain any more information from load builds than they already know. As a result more load builds increase development time. In fact one project leader stated that he would have improved his development process by having fewer builds of the product, allowing for more full system tests. Further research is necessary to determine which development situations lend themselves to running more load builds and which situations do not.

With respect to testing, while a positive relationship was found between emphasis on unit testing and software development time, both the number of

system tests and beta test sites were negatively correlated with software development time. An explanation of the positive relationship between emphasis on unit testing and software development time may be that engineers become too preoccupied with continuously adding increased functionality in their own code rather than quickly arriving at a complete system which could then be tested as a whole. In a sense, engineers may become too attached with their own piece of to product and lose sight of the overall project goal.

Tied with the above are the results that indicate that more system tests and beta test sites reduce software development time. This follows Gupta and Wilemon's (1990) belief that frequent product testing shortens software development time because problems are discovered earlier in the process when they are easier to rectify. This also supports the Eisenhardt and Tabrizi (1995) hypothesis that when a company is not familiar with the development of a product, frequent testing reduces development time. A more comprehensive study into product testing during software development is required to determine what testing procedures are of most benefit software development, which development situations are best suited for the use of these procedures, and how much emphasis should be placed on them during each situation.

Project leader control over product specifications and annual evaluations of development personnel were both found to be negatively correlated with software development time. If control over product specifications and annual evaluations of development personnel can be accepted as indicators of power, this result supports the Bowen et. al (1994) notion that more project leader power is associated with shorter development time. As was the case here, this is especially true of low experience companies. An explanation of this may be that when a company lacks experience developing a product it is very important to have one strong leader with a clear vision of what the product should be. If this does not exist, the product may be pulled in different directions resulting in a longer development process. More research into the different dimensions of project leader power in software development would provide more insight into which aspects of the development process project leaders should be given more control over.

The qualitative questions which asked project leaders which aspects of their development process they would like more control over and how they would improve their development processes provide an excellent basis for further investigation into project leader power. The aspect which is most interesting about these questions is that areas of development over which project leaders would like more control were not necessarily areas which they

would improve. For example, project leaders indicated that they would improve product testing, but they did not indicate that they would like more control over that aspect of the development process. The same applies for training of development personnel. Project leaders also indicated that they would like more control over development resources and time frames, but fewer indicated that they would improve these aspects of development. It would be very interesting to see how the dimensions of project leader control identified in these questions relate to software development time. To investigate project leader control future research could discover the extent project leaders have control over development resources (financial and personnel selection), development time frames, and the defining of product requirements. This could be accomplished easily just by preparing a series of Likert style questions based on these dimensions.

On a number of occasions it was stated that the residuals were correlated with software development time, indicating an explanatory variable is missing. More research is required to determine what these explanatory variables are. Some clues may be found in the question that asked project leaders how they would improve their development process. The leading areas mentioned were better product testing and more end user/customer involvement. A more in depth study is needed into software testing

procedures. What procedures best lend themselves to certain development situations must be determined and tested against software development time.

Research is also needed into the effect of customer/end user involvement in the development process on development time. Also, the effects of giving project leaders more control over development resources, time frames and product specifications must be tested to determine whether this may affect development time. Within these areas may lie the answer to the mystery of the missing explanatory variables.

5.2 Insights for Managers

Academic results are useless if they do not offer insight into improving actual processes and managerial practices. The results of this study can definitely accomplish this. The overwhelming insight that this research provides is that managers must consider the company's prior experience in developing a product and stress different development aspects accordingly. For low experience companies, it has been shown that supplier involvement reduces software development time. Managers must ensure that their companies maintain close relationships with suppliers to keep abreast of new development tools they are producing. Ideally companies could discuss future

projects with their suppliers such that suppliers could recommend or possibly develop new tools to speed development.

Another managerial insight from the research is that functional involvement during coding slows development. Possibly, functional involvement should be limited to the early development activities especially when a company has considerable experience developing a product. In fact the results indicate that when little experience exists, involving sales and marketing in defining product requirements may shorten development time. Once product requirements and specifications are set however, software engineers should be left to create the full system. Tied to this is the result that frequent load builds slow development time. When experience is high the frequency of load builds should be limited to ensure a full system can be built and tested as soon as possible. More system tests would encourage more functional involvement because a more tangible product would offer people more opportunities to try the product and offer immediate feedback.

The research results indicate that more system tests and beta test sites quicken the pace of development. This result is even stronger for low experience projects. This reinforces the idea that a full working system should be established as early into development as possible. Managers should encourage the use of beta test sites to encourage customer involvement in

the development process. Once again, the qualitative results show that this, along with product testing are the two leading aspects that project leaders would improve in their development processes. Having more beta test sites may be a step towards improving both of these aspects.

Both the quantitative and qualitative results demonstrate that project leaders should definitely be given more control over product specifications and annual evaluations of development personnel. Project leaders should also be allowed more control over development resources and time frames. They should also be more active in the defining product requirements. This will provide them with a stronger sense of commitment and ownership to the product. As a result, the development process will run smoother because workers will have a clearer understanding of who is in control.

5.3 Limitations of Study

The major limitation of this study is generalizability. Not all of the results from this research will be generalizable because of the limited sample size used. Also, since there was one respondent for each project, many of the responses are bound to be subjective. Data reliability might have been improved by using more than one respondent. However, since obtaining completed questionnaires was a time and cost intensive exercise, this would

have meant sampling a smaller number of projects.

The researcher also acknowledges the fact that although direct observation is probably the best means to evaluate a project, the nature of the software development process makes direct observation difficult. A longitudinal research design that follows an entire development project from start to finish may have also enabled the researcher to examine the other aspects of software development. However, once again, time and cost constraints did not permit this.

References

- Boehm, B. W. (1976) "Software Engineering Economics," IEEE Transactions on Computers, 25(12): 1226-1241.
- Boehm, B. W. (1988) "A Spiral Model of Software Development and Enhancement," IEEE Transactions on Computers, 21(5): 61-72.
- Bowen, Kent H., Kim B. Clark, Charles A. Holloway, and Steven C. Wheelwright (1994) The Perpetual Enterprise: Seven keys to corporate Renewal through successful product and process development, Oxford University Press (New York).
- Clark, Kim B. and T. Fujimoto (1991) Product Development Performance, Harvard Business School Press (Boston).
- Clark, Kim B. and Steven C. Wheelwright (1992) "Organizing and Leading Heavyweight Development Teams," California Management Review, 34(3): 9-28.
- Cusumano, Michael and Richard W. Shelby (1995) Microsoft Secrets, The Free Press (New York).
- Eisenhardt, Kathleen M. (1989) "Making Fast Strategic Decisions in High Velocity Environments," Academy of Management Journal, 32: 543-576.
- Eisenhardt, Kathleen M. and Behnam N. Tabrizi (1995) "Adaptive Processes: Product Innovation in the global computer industry," Administrative Science Quarterly, 40: 46-70.
- Feldman, Martha S., and James G. March (1981) "Information in Organizations as Signals and Symbols," Administrative Science Quarterly, 26: 171-186.
- Gupta, Ashok K., and David L. Wilemon (1990) "Accelerating the Development of Technology-based New Products," California Management Review, 32(2): 29-44.

Imai, Kenochi, Ikujiro Nonaka, and Hirotaka Takeuchi (1985) "Managing the New Product Development Process: How Japanese Companies Learn and Unlearn," In R. Hayes, K. Clark, and P. Lorenz (eds.), The Uneasy Alliance: Managing the Productivity-Technology Dilemma, 337-375. Harvard Business School Press (Boston).

Mabert, Vincent A., John F. Muth, and Roger W. Schmenner (1992) "Collapsing New Product Development Times: Six Case Studies," Journal of Product Innovation Management, 9: 200-212.

Pfeffer, Jeffrey (1992) "Managing Innovation: Controlled Chaos," Harvard Business Review, May-June: 73-84.

Zirger, B. J. and Janet L. Hartley, (1996) "The Effect of Acceleration Techniques on Product Development Time " IEEE Transactions on Engineering Management, 43(2): 143-151.

Appendix A - Questionnaire

****To respond to this survey, create a reply e-mail message that contains the survey. Some e-mail systems require you to manually copy and paste the survey into your reply.**

****Please make sure the reply contains the survey authentication marker.**

****To answer a question, type an x between the brackets, like this: [x]. For fill-in-the-blanks, type between the brackets like this: [your response]. Please make no other changes to this survey.**

****Please select a software development project with which you were involved within the three years ending December 1st of last year and answer the following questions. If the project involved more than developing a software product (e.g. developing a hardware product), please focus on the software development aspects of the project and the team responsible for developing the software product.**

****This questionnaire should take less than 30 minutes to complete.**

****ALL RESPONSES WILL BE KEPT CONFIDENTIAL.**

****In the case of missing responses, I may need to contact you personally. Also, if you wish I will send you a copy of my research results. Please provide your:**

Name:

Email:

Phone:

****Place an "X" here ____ if you wish to receive a copy of research results**

Thank you for your time and cooperation.

THE PRODUCT AND THE PROCESS USED:

1. Please indicate the date personnel were first assigned to the project to analyze product requirements.

Enter date (dd/mm/yy) []

2. Please indicate the date when the commercial specification for this product was finalized.

Enter date (dd/mm/yy) []

3. Please indicate the date when source coding for the product began.

Enter date (dd/mm/yy) []

4. Please indicate the date the product was released to the end user/customer.

Enter date (dd/mm/yy) []

5. Rate the organization's prior experience in developing such a product?

Choose ONE (place an X in the appropriate bracket)

- [] a) 1- None**
- [] b) 2**
- [] c) 3**
- [] d) 4- Some**
- [] e) 5**
- [] f) 6**
- [] g) 7- Considerable**

6. Rate how similar was the product developed to other products that have been developed by your organization?

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- Very Different
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Similar
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Very Similar

7. Rate how similar was the development process for this project to past projects undertaken by your organization?

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- Very Different
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Similar
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Very Similar

8. Were any new software development tools or processes used to develop the new product?

Choose one:

- ☐ a) Yes
- ☐ b) No

9. Rate how important was the development of the product to the success of the firm?

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- Not Important
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Somewhat Important
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Very Important

SUPPLIER INVOLVEMENT IN THE DEVELOPMENT PROCESS:

10. Using the following definitions:

*defining product requirements: is the determination of the commercial or customer requirements for the product together with its functional specification

*system design: is the definition of the product design at the level of the modules to be used and their interdependencies

*creation of source code: is exactly that - the creation of the source code for the software product in whatever languages that are used

*system testing: is the internal, laboratory type testing of version's of the whole product

*beta testing: is the testing of pre-release versions of the product by an end user/customer often on their site

INDICATE WITH AN "X" whether one or more of a third party supplier's personnel actively participated in the development activities defined below:

Choose all that apply:

- ☐ a) Defining product requirements
- ☐ b) System design
- ☐ c) Creation of source code
- ☐ d) System testing
- ☐ e) Beta testing
- ☐ f) Not applicable

11. Rate how important the third party supplier's personnel were in helping to make key decisions on issues such as resource allocation during the DEFINITION OF PRODUCT REQUIREMENTS.

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- Not Important
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Somewhat Important
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Very Important

12. Rate how important the third party supplier's personnel were in helping to make key decisions on issues such as resource allocation during SYSTEM DESIGN.

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- Not Important
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Somewhat Important
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Very Important

13. Rate how important the third party supplier's personnel were in helping to make key decisions on issues such as resource allocation during the CREATION OF SOURCE CODE.

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- Not Important
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Somewhat Important
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Very Important

14. Rate how important the third party supplier's personnel were in helping to make key decisions on issues such as resource allocation during SYSTEM TESTING.

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- Not Important
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Somewhat Important
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Very Important

15. Rate how important the third party supplier's personnel were in helping to make key decisions on issues such as resource allocation during BETA TESTING.

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- Not Important
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Somewhat Important
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Very Important

FUNCTIONAL INVOLVEMENT:

16. What was the "head count" in person months initially allocated to this project?

Enter number of person months (e.g., 500) ☐

17. What was the total cost the project?

Enter dollar amount (e.g., 10,000) ☐

18. Indicate with an "X" whether personnel from the company's MARKETING AND SALES department helped to make key decisions (on issues such as resource allocation, product specifications, and system design) during each of the following activities in the development process.

Choose all that apply:

- ☐ a) Defining product requirements
- ☐ b) System design
- ☐ c) Creation of source code
- ☐ d) System testing
- ☐ e) Beta testing
- ☐ f) Not applicable

19. Indicate with an "X" whether personnel from PRODUCT MANAGEMENT helped to make key decisions (on issues such as resource allocation, product specifications, and system design) during each of the following activities in the development process.

Choose all that apply:

- ☐ a) Defining product requirements
- ☐ b) System design
- ☐ c) Creation of source code
- ☐ d) System testing
- ☐ e) Beta testing
- ☐ f) Not applicable

20. Indicate with an "X" whether personnel from the company's CUSTOMER SERVICES department helped to make key decisions (on issues such as resource allocation, product specifications, and system design) during each of the following activities in the development process.

Choose all that apply:

- ☐ a) Defining product requirements
- ☐ b) System design
- ☐ c) Creation of source code
- ☐ d) System testing
- ☐ e) Beta testing
- ☐ f) Not applicable

21. Indicate with an "X" whether personnel from the company's QUALITY ASSURANCE department helped to make key decisions (on issues such as resource allocation, product specifications, and system design) during each of the following activities in the development process.

Choose all that apply:

- ☐ a) Defining product requirements
- ☐ b) System design
- ☐ c) Creation of source code
- ☐ d) System testing
- ☐ e) Beta testing
- ☐ f) Not applicable

22. Indicate with an "X" whether personnel from the company's SOFTWARE MAINTENANCE department helped to make key decisions (on issues such as resource allocation, product specifications, and system design) during each of the following activities in the development process.

Choose all that apply:

- ☐ a) Defining product requirements
- ☐ b) System design
- ☐ c) Creation of source code
- ☐ d) System testing
- ☐ e) Beta testing
- ☐ f) Not applicable

23. Indicate with an "X" whether personnel from the company's SOFTWARE DESIGN department helped to make key decisions (on issues such as resource allocation, product specifications, and system design) during each of the following activities in the development process.

Choose all that apply:

- ☐ a) Defining product requirements
- ☐ b) System design
- ☐ c) Creation of source code
- ☐ d) System testing
- ☐ e) Beta testing
- ☐ f) Not applicable

REWARDS:

24. For this project, what was the total expense in the form of base pay to project personnel?

Enter total dollar amount of base pay expense (e.g., 100,000) []

25. What was the total expense for schedule attainment performance bonuses to project personnel over and above their base pay?

Enter total dollar amount of performance bonus expense (e.g., 20,000) []

LOAD BUILDS:

26. How many load builds of the product were carried out throughout the project?

Enter number of load builds []

TESTING:

27. How many times was the full system tested during the project?

Enter number of full system tests []

28. How many beta test sites were used in the development of the product?

Enter number of beta test sites []

29. Rate how important was unit testing to the overall testing of the product?

Choose ONE (place an X in the appropriate bracket)

- [] a) 1- Not Important
- [] b) 2
- [] c) 3
- [] d) 4- Somewhat Important
- [] e) 5
- [] f) 6
- [] g) 7- Very Important

PROJECT LEADER:

30. Rate your own decision making authority over product specifications in this project.

Choose ONE (place an X in the appropriate bracket)

- [] a) 1- None
- [] b) 2
- [] c) 3
- [] d) 4- Some
- [] e) 5
- [] f) 6
- [] g) 7- Final Decision Maker

31. Rate your own authority over annual evaluations of development personnel.

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- None
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Some
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Final Decision Maker

32. Name some aspects of this software development project that you would of liked more control over.

Enter text within the brackets []

GENERAL QUESTION:

33. How would you have improved this development project?

Enter text within the brackets []

Thank you for your time and cooperation.

Since I am a student with a disability who has a speech impediment, you may have difficulties understanding me over the phone. If you have any questions you can email me at bmoretto@ccs.carleton.ca.

For those who indicated interest, I will send a copy of my research results as soon as the thesis is completed.

Appendix B - Mann-Whitney U Comparisons of Means

Development time

Mean Rank	Sum of Ranks	Cases
17.58	351.5	20 High Experience
20.68	351.5	17 Low Experience
37 Total		

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
141.5	351.5	.3896	-.8687	.3850

**This exact p-value is not corrected for ties.

Project cost

Mean Rank	Sum of Ranks	Cases
20.70	414.0	20 High Experience
17.00	289.0	17 Low Experience
37 Total		

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
136.0	289.0	.3113	-1.0370	.2997

**This exact p-value is not corrected for ties.

TESTING:

27. How many times was the full system tested during the project?

Enter number of full system tests []

28. How many beta test sites were used in the development of the product?

Enter number of beta test sites []

29. Rate how important was unit testing to the overall testing of the product?

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- Not Important
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Somewhat Important
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Very Important

PROJECT LEADER:

30. Rate your own decision making authority over product specifications in this project.

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- None
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Some
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Final Decision Maker

31. Rate your own authority over annual evaluations of development personnel.

Choose ONE (place an X in the appropriate bracket)

- ☐ a) 1- None
- ☐ b) 2
- ☐ c) 3
- ☐ d) 4- Some
- ☐ e) 5
- ☐ f) 6
- ☐ g) 7- Final Decision Maker

32. Name some aspects of this software development project that you would of liked more control over.

Enter text within the brackets []

GENERAL QUESTION:

33. How would you have improved this development project?

Enter text within the brackets []

Thank you for your time and cooperation.

Since I am a student with a disability who has a speech impediment, you may have difficulties understanding me over the phone. If you have any questions you can email me at bmoretto@ccs.carleton.ca.

For those who indicated interest, I will send a copy of my research results as soon as the thesis is completed.

Functional involvement in defining product requirements

Mean Rank	Sum of Ranks	Cases
23.30	466.0	20 High Experience
13.94	237.0	17 Low Experience

37 Total

Exact**				
U	W	(One-Tailed P)	Z	2-Tailed P
84.0	237.0	.0080	-2.6834	.0073

**This exact p-value is not corrected for ties.

Functional involvement in system design

Mean Rank	Sum of Ranks	Cases
19.23	384.5	20 High Experience
18.74	318.5	17 Low Experience

37 Total

Exact**				
U	W	(One-Tailed P)	Z	2-Tailed P
165.5	318.5	.8923	-.1418	.8872

**This exact p-value is not corrected for ties.

Functional involvement in coding

Mean Rank	Sum of Ranks	Cases
19.70	394.0	20 High Experience
18.18	309.0	17 Low Experience

37 Total

Exact**				
U	W	(One-Tailed P)	Z	2-Tailed P
156.0	309.0	.6843	-.4460	.6556

**This exact p-value is not corrected for ties.

Functional involvement in system testing

Mean Rank	Sum of Ranks	Cases
21.27	425.5	20 High Experience
16.32	277.5	17 Low Experience

37 Total

Exact**				
U	W	(One-Tailed P)	Z	2-Tailed P
124.5	277.5	.1675	-1.4478	.1477

**This exact p-value is not corrected for ties.

Functional involvement in beta testing

Mean Rank	Sum of Ranks	Cases
20.55	411.0	20 High Experience
17.18	292.0	17 Low Experience

37 Total

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
139.0	292.0	.3569	-.9649	.3346

**This exact p-value is not corrected for ties.

Financial rewards for schedule attainment

Mean Rank	Sum of Ranks	Cases
18.60	372.0	20 High Experience
19.47	331.0	17 Low Experience

37 Total

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
162.0	372.0	.8214	-.2859	.7750

**This exact p-value is not corrected for ties.

Number of load builds

Mean Rank	Sum of Ranks	Cases
22.73	454.5	20 High Experience
14.62	248.5	17 Low Experience

37 Total

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
95.5	248.5	.0219	-2.2737	.0230

**This exact p-value is not corrected for ties.

Number of unit tests

Mean Rank	Sum of Ranks	Cases
19.08	381.5	20 High Experience
18.91	321.5	17 Low Experience

37 Total

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
168.5	321.5	.9640	-.0482	.9616

**This exact p-value is not corrected for ties.

Number of system tests

Mean Rank	Sum of Ranks	Cases
20.30	406.0	20 High Experience
17.47	297.0	17 Low Experience

37 Total

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
144.0	297.0	.4417	-.7975	.4252

**This exact p-value is not corrected for ties.

Number of beta sites

Mean Rank	Sum of Ranks	Cases
19.77	395.5	20 High Experience
18.09	307.5	17 Low Experience

37 Total

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
154.5	307.5	.6405	-.4759	.6342

**This exact p-value is not corrected for ties.

Project leader control over product specifications

Mean Rank	Sum of Ranks	Cases
17.73	354.5	20 High Experience
20.50	348.5	17 Low Experience

37 Total

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
144.5	354.5	.4417	-.8016	.4228

**This exact p-value is not corrected for ties.

Project leader annual evaluations of development personnel

Mean Rank	Sum of Ranks	Cases
20.65	413.0	20 High Experience
17.06	290.0	17 Low Experience

37 Total

U	W	Exact** (One-Tailed P)	Z	2-Tailed P
137.0	290.0	.3261	-1.0790	.2806

**This exact p-value is not corrected for ties.

Appendix C - Definition of Variables

defproreq: proportion of time spent defining product requirements (Number of weeks defining product requirements / software development time)

q5expr: company's prior experience developing a product

q6prod: how similar the product was to previously developed products

q7process: how similar the development process was to previously developed products

q8ntools: whether any new software development tools or processes were used

q9imp: importance of the product to success of firm

q10suppa: supplier involvement in defining product requirements

q10suppb: supplier involvement in system design

q10suppc: supplier involvement in coding

q10suppd: supplier involvement in system design

q10suppe: supplier involvement in beta testing

q10suppf: not applicable

q11simpa: supplier involvement importance in defining product requirements

q12simpb: supplier involvement importance in system design

q13simpc: supplier involvement importance in coding

q14simpd: supplier involvement importance in system design

q15simpe: supplier involvement importance in beta testing

q16simpf: not applicable

supinv: supplier involvement total

mktgdpr: sales and marketing involvement in defining product requirements

mktgsd: sales and marketing involvement in system design

mktgcod: sales and marketing involvement in coding
mktgst: sales and marketing involvement in system testing
mktgbt: sales and marketing involvement in beta testing
mktgna: not applicable
pmdpr: product management involvement in defining product requirements
pmsd: product management involvement in system design
pmcod: product management involvement in coding
pmst: product management involvement in system testing
pmbt: product management involvement in beta testing
pmna: not applicable
cssdpr: customer services involvement in defining product requirements
cssd: customer services involvement in system design
cscod: customer services involvement in coding
csst: customer services involvement in system testing
csbt: customer services involvement in beta testing
csna: not applicable
qadpr: quality control involvement in defining product requirements
qasd: quality control involvement in system design
qacod: quality control involvement in coding
qast: quality control involvement in system testing
qabt: quality control involvement in beta testing
qana: not applicable
smdpr: software maintenance involvement in defining product requirements
smsd: software maintenance involvement in system design
smcod: software maintenance involvement in coding
smst: software maintenance involvement in system testing

smbt: software maintenance involvement in beta testing
smna: not applicable
sddpr: software design involvement in defining product requirements
sdsd: software design involvement in system design
sdcod: software design involvement in coding
sdst: software design involvement in system testing
sdbt: software design involvement in beta testing
sdna: not applicable
funinv: total functional involvement
unit: importance/emphasis of unit testing
pmspec: project leader control over product specifications
pmevals: project leader control over annual evaluations of development
personnel
bonus: performance bonus for schedule attainment
builds: number of load builds
systests: number of full system tests
betasite: number of beta test sites
head#: initial head count
cost: total cost of project
bpayex: base pay expense
bonuspc: bonus as a % of base pay expense
start: project start date
comspec: date commercial spec was finalized
coding: date coding began
release: date product was released
dplength: total development time from start to release

prodreq: time spent defining product requirements (start to coding)
dpnorm: (total development time from start to release * 100,000)/cost
prodnorm: (time spent defining product requirements (start to coding))/cost
rest: development time minus time spent defining product requirements
(dplength - prodreq)

IMAGE EVALUATION TEST TARGET (QA-3)

APPLIED IMAGE, Inc.
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved