1

**PM-1 3 1/2" x 4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT**

PRECISIONSM RESOLUTION TARGETS

**AN INVESTIGATION OF THE DIMENSIONS OF PROTOTYPE
MANAGEMENT THAT AFFECT THE OUTCOMES OF SOFTWARE
DEVELOPMENT PROJECTS**

**by
FRASER Mac CONNELL SMITH**

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

Master of Management Studies

School of Business

Carleton University

Ottawa, Ontario

September 1996

© Copyright

1996, Fraser Smith

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa (Ontario)
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395 rue Wellington
Ottawa (Ontario)
K1A 0N4

pour les usages de référence

pour les usages de référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

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

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

ISBN 0-612-19533-3

Canada

Name Flora E. Council Smith

Dissertation Abstracts International and Masters Abstracts International are arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation or thesis. Enter the corresponding four-digit code in the spaces provided.

SUBJECT TERM

UMI
SUBJECT CODE

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
Art History 0377
Classics 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0499
Theater 0465

EDUCATION

General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0488
Community College 0275
Curriculum and Instruction 0277
Early Childhood 0518
Elementary 0524
French 0277
Guidance and Counseling 0519
Health 0480
Higher 0745
History of 0520
Home Economics 0278
Industrial 0421
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523

Psychology 0525
Reading 0535
Religion 0527
Sciences 0714
Secondary 052
Social Sciences 0534
Sociology of 0140
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language
General 0679
Ancient 0289
Linguistics 0290
Modern 0291
Literature
General 0401
Classical 0294
Comparative 0295
Medieval 0297
Modern 0298
African 0316
American 0317
Asian 0305
Canadian (English) 0352
Canadian (French) 0353
English 0393
Germanic 0311
Latin American 0312
Middle Eastern 0315
Romance 0313
Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
Religion
General 0318
Biblical Studies 0321
Clergy 0319
History of 0320
Philosophy of 0322
Theology 0469

SOCIAL SCIENCES

American Studies 0323
Anthropology
Archaeology 0324
Cultural 0326
Physical 0327
Business Administration
General 0310
Accounting 0272
Banking 0770
Management 0454
Marketing 0338
Canadian Studies 0385
Economics
General 0501
Agricultural 0503
Commerce-Business 0505
Finance 0508
History 0509
Labor 0510
Theory 0511
Polls 0558
Geography 0346
Genealogy 0351
History
General 0578

Ancient 0579
Medieval 0581
Modern 0582
Black 0328
African 07
Asia, Australia and Oceania 0332
Canadian 0334
European 0335
Latin American 0336
Middle Eastern 0333
United States 0337
History of Science 0585
Law 0398
Political Science
General 0415
International Law and
Relations 0414
Public Administration 0417
Recreation 0814
Social Work 0452
Sociology
General 0426
Criminology and Penology 0427
Demography 0938
Ethnic and Racial Studies 0431
Individual and Family
Studies 0438
Industrial and Labor
Relations 0429
Public and Social Welfare 0430
Social Structure and
Development 0700
Theory and Methods 0344
Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture 0473
General 0285
Agronomy 0475
Animal Culture and
Nutrition 0476
Animal Pathology 0476
Food Science and
Technology 0356
Forest and Wildlife 0478
Plant Culture 0479
Plant Pathology 0480
Plant Physiology 0481
Range Management 0777
Wood Technology 0744

Biology

General 0306
Anatomy 0287
Botanics 0308
Botany 0309
Cell 0379
Ecology 0329
Evolutionary 0353
Genetics 0349
Immunology 0793
Microbiology 0410
Molecular 0307
Neuroscience 0317
Oncogenetics 0414
Physiology 0433
Radiation 0421
Veterinary Science 0778
Zoology 0472

EARTH SCIENCES

Biogeochemistry 0425
Geochemistry 0996

Geology 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoclimatology 0426
Paleontology 0418
Paleozoology 0985
Physical Geography 0348
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0748
Health Sciences
General 0544
Anatomy 0300
Chemotherapy 0972
Dentistry 0547
Education 0390
Hospital Management 0749
Human Development 0738
Immunology 0982
Medicine and Surgery 0544
Mental Health 0347
Nursing 0549
Nutrition 0570
Obstetrics and Gynecology 0380
Occupational Health and
Safety 0354
Ophthalmology 0381
Pathology 0571
Pharmacology 0419
Pharmacy 0572
Physical Therapy 0382
Public Health 0573
Radiology 0574
Recreation 0575

Speech Pathology 0440
Toxicology 0383
Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences
Chemistry
General 0485
Agricultural 0749
Analytical 0486
Biochemistry 0487
Inorganic 0488
Nuclear 0738
Organic 0490
Pharmaceutical 0491
Physical 0494
Polymer 0495
Radiation 0784
Mathematics 0405
Physics
General 0406
Acoustics 0994
Astronomy and
Astrophysics 0404
Atmospheric Science 0408
Atomic 0748
Electronic and Electrical 0407
Elementary Particles and
High Energy 0798
Field and Plasma 0799
Molecular 0409
Nuclear 0410
Optics 0782
Radiation 0784
Solid State 0411
Statistics 3453
Applied Sciences
Applied Mechanics 0346
Computer Science 0984

Engineering

General 0537
Aerospace 0538
Agricultural 0539
Automotive 0540
Biomedical 0541
Chemical 0542
Civil 0543
Electronic and Electrical 0544
Heat and Thermodynamics 0545
Hydraulic 0546
Industrial 0547
Marine 0548
Materials Science 0549
Mechanical 0550
Metallurgy 0551
Mining 0552
Packaging 0553
Petroleum 0554
Sanitary and Municipal
System Science 0555
Geotechnology 0556
Operations Research 0557
Plastic Technology 0558
Textile Technology 0559

PSYCHOLOGY

General 0421
Behavioral 0384
Child 0422
Developmental 0423
Experimental 0424
Industrial 0425
Personality 0426
Physiological 0427
Psychiatry 0428
Social 0429

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

submitted by Fraser Mac Connell Smith, B. Bus. Admin. (Honours).
in partial fulfillment of the requirements for the degree of Master of Management Studies

Thesis Supervisor

Director, School of Business

Carleton University

September 20, 1996

To Mum and Dad

Here's tae us
Wha's like us
Die'l the yin
an' their a' deid

**AN INVESTIGATION OF THE DIMENSIONS OF PROTOTYPE
MANAGEMENT THAT AFFECT THE OUTCOMES OF SOFTWARE
DEVELOPMENT PROJECTS**

by

Fraser Mac Connell Smith

ABSTRACT

The objective of this research is to examine how prototype management practice affects three software project outcomes: project duration, schedule overrun and project manager's satisfaction that the software met customer requirements. Seven hypotheses are tested using data from 40 software development projects drawn from companies located in Canada.

The results show that project outcomes are not linked to the number of prototypes used throughout the development process, external involvement, the number of customer sites used to assess prototypes and the effort dedicated to prototyping.

Project duration was found to be affected by: 1. the extent to which computer scientists and engineers are involved in both software development and prototype development, 2. functional involvement throughout the development process, 3. functional involvement at the completion of the software's high level design and 4. the extent to which

the prototype embodies the characteristics of the end-product at the completion of the software's high level design.

The results suggest that schedule overrun is associated with the extent of functional involvement in prototype development which occurs after completion of the software's high level design. The greater the number of internal functions involved in prototyping during the later stages of software development, the greater the schedule overrun.

Project manager's satisfaction that the software met customer needs was found to be associated with the extent to which the prototype embodied the characteristics of the end-product at the completion of the software's high level design.

There are a variety of tools and languages that are used to prototype. Most of these languages are programming languages that implement the object-oriented approach to development.

The decisions that contributed to shorten project duration, reduce schedule overrun and increase the project manager's satisfaction that the software met customer requirements can be grouped into five categories: i. timely resource allocation, ii. incorporation of stakeholder's concerns into the design, iii. development process characteristics, iv. maintenance of schedule and product vision and v. experience of team members.

ACKNOWLEDGMENTS

I thank the School of Business for providing me with the opportunity to conduct this research.

I am extremely thankful to Prof. Antonio J. Bailetti, my thesis supervisor, for his unwavering interest, financial support, invaluable comments, patience and guidance throughout the design, development and prototyping of this thesis. Further, Prof. Bailetti provided me with a learning environment that greatly expanded my skills and knowledge.

I thank Professor John Callahan for his continued interest, invaluable comments and insights.

I thank Professor Debbie Compeau for her comments and suggestions.

I thank Professor Shirley Mills for her invaluable guidance and assistance with the statistical analysis.

I thank Professor Dorina Petriu for her continued interest and invaluable comments from a software engineering perspective.

The time and effort of all the presidents, vice presidents, project managers, team leaders and software engineers who provided the data and suggestions are gratefully acknowledged.

I am grateful to Professors Barbara Austin, Tansu Barker, and Peter Yannopoulos of Brock University, Canada, for encouraging me to pursue a Master's degree.

Ms. Jean Blair's kindness and efforts to accommodate me is much appreciated.

I thank my parents, Agnes and John Mac Connell Smith, for their encouragement, guidance and support through the trials and tribulations of this thesis.

TABLE OF CONTENTS

Acceptance Sheet	ii
Abstract	iv
Acknowledgments	vi
Table of Contents	vii
List of Tables	x
List of Figures	xii
List of Appendices	xiii
 Chapter 1 Introduction	 1
1.1 Objective.....	2
1.2 Prototype, Software Prototype and Prototype Process.....	2
1.3 Contribution to Academic Literature.....	4
1.4 Contribution to Management Practice.....	4
1.5 Organization of the Thesis.....	5
 Chapter 2 Literature Review and Hypotheses	 6
2.1 Classification of Software Prototypes.....	6
2.2 Project Outcomes.....	9
2.3 Stages of Software Development.....	11
2.4 Testable Hypotheses.....	12
2.5 Summary of Hypotheses.....	24
2.6 Aspects of Prototype Management Identified in the Literature that Were Not Examined in this Thesis.....	25

Chapter 3 Research Design	29
3.1 Unit of Analysis	29
3.2 Sample	29
3.3 Company, Project and Respondent Selection	29
3.4 Time of Data Collection	30
3.5 Data Collection Plan	30
3.5.1 Piloting the Questionnaire	30
3.5.2 Data Collection	31
3.6 Measures	31
3.7 Data Analysis	36
Chapter 4 Results	38
4.1 Response Rate and Breakdown	38
4.2 Tools and Languages Used to Prototype	39
4.3 Independent Variables	41
4.4 Dependent Variables	45
4.5 Project Size	46
4.6 Exploratory Data Analysis	47
4.7 Results of Regression Analyses for Project Duration and Schedule Overrun	49
4.8 Results of Correlation Analyses for Project Manager's Satisfaction that the Software Met Customer Requirements	75
4.9 Summary of Significant Results from Ordinary Least Squares Regression and Zero-Order Correlations	77
4.10 Results of Stepwise Regression Analyses for Project Duration and Schedule Overrun	79

4.11	List of Important Prototype Management Decisions Made to Shorten Project Duration, Schedule Overrun and Increase Project Manager's Satisfaction that the Software Met Customer Requirements.....	83
4.11.1	Important Prototype Management Decisions that Contributed to a Reduction in Project Duration.....	83
4.11.2	Important Prototype Management Decisions that Contributed to a Reduction in Schedule Overrun.....	85
4.11.3	Important Prototype Management Decisions that Contributed to An Increase in the Project Manager's Satisfaction that the Software Met Customer Requirements	87
Chapter 5	Discussion of Results.....	90
5.1	Discussion of Results.....	90
5.2	Integrate Software and Prototype Development Teams.....	92
5.3	Involve Functions in Prototyping Early	92
5.4	Extent that the Product is Embodied in the Prototype	93
5.5	Use Prototypes to Document and Track the Progress of the Project	94
5.6	Prototyping Tools and Languages.....	96
5.7	Limitations of Research Approach	97
References		98

LIST OF TABLES

Table	Description	Page
1	Aspects of Prototype Management Identified in the Software and Product Development Literature Examined in the Thesis	26
2a	Tools Used to Prototype	40
2b	Languages Used to Prototype	41
3	Range, Means, Modes, Standard Deviations and Number of Observations for Independent Variables	44
4	Range, Means, Modes, Standard Deviations and Number of Observations for Dependent Variables	45
5	Correlations for Project Duration, Schedule Overrun and Project Manager's Satisfaction that the Software Met Customer Requirements	46
6	Range, Means, Modes, Standard Deviations and Number of Observations for the Variables Used to Measure Project Size	46
7	Correlations for Total Code, Total Cost and Total person-months	47
8	Results of Forward Selection Regression Analysis	48
9	Results of Regression Analysis for Hypothesis 1	50
10	Results of Regression Analysis for Evolutionary Prototypes	51
11	Results of Regression Analysis for Hypothesis 2	52
12	Results of Regression Analysis for Hypothesis 3 (Across All Stages)	54
13	Results of Regression Analysis for Hypothesis 3 (Stage 1)	55
14	Results of Regression Analysis for Hypothesis 3 (Stage 2)	56
15	Results of Regression Analysis for Hypothesis 3 (Stage 3)	57
16	Results of Regression Analysis for Hypothesis 3 (Stage 4)	58

17	Results of Regression Analysis for Hypothesis 4 (Across All Stages)	60
18	Results of Regression Analysis for Hypothesis 4 (Stage 1)	61
19	Results of Regression Analysis for Hypothesis 4 (Stage 2)	62
20	Results of Regression Analysis for Hypothesis 4 (Stage 3)	63
21	Results of Regression Analysis for Hypothesis 4 (Stage 4)	64
22	Results of Regression Analysis for Hypothesis 5	65
23	Results of Regression Analysis for Hypothesis 6 (Across All Stages)	66
24	Results of Regression Analysis for Hypothesis 6a (Stage 1)	68
25	Results of Regression Analysis for Hypothesis 6b (Stage 2)	69
26	Results of Regression Analysis for Hypothesis 6c (Stage 3)	70
27	Results of Regression Analysis for Hypothesis 6d (Stage 4)	71
28	Results of Regression Analysis for Hypothesis 7a	72
29	Results of Regression Analysis for Hypothesis 7b	73
30	Results of Regression Analysis for Hypothesis 7c	74
31	Correlations for Project Manager's Satisfaction that the Software Met Customer Requirements	76
32	Summary of Significant Results From Ordinary Least Squares Regression and Zero-Order Correlations	78
33	Results of Stepwise Regression Analysis for Project Duration	80
34	Results of Stepwise Regression Analysis for Schedule Overrun	81

LIST OF FIGURES

Figure	Description	Page
1	The Software Development Process	12

LIST OF APPENDICES

Appendix	Title	Page
A	Cover Letter Sent to Presidents or Vice Presidents of Selected Companies	101
B	Cover Letter Sent To Manager of Project Selected	102
C	Prototype Management Questionnaire	103
D	Correlations for Independent Variables	108

CHAPTER 1

INTRODUCTION

Prototyping is a major area of opportunity for managers who seek to improve the effectiveness and efficiency of their development processes (Wheelwright & Clark, 1992: 664). Barkin and Iansiti (1994: 19) identified the effective and timely use of prototypes as one of seven key elements for outstanding development projects.

Researchers associate prototyping with various project benefits. It has been suggested that prototyping can be used to reduce the cost of projects and shorten project duration (Barkin & Iansiti, 1994: 204; Längle, Leitheiser & Naumann, 1984: 273), to identify engineering problems early before engineering changes become expensive (Clark & Fujimoto, 1991: 119 and Ulrich & Eppinger, 1995: 224), to reduce development risk (Bischofberger & Pomberger, 1992: 3 and Barkin & Iansiti, 1994: 204), to improve the accuracy of the development process (Lichter, Schneider-Hufschmidt & Züllighoven, 1994: 825), to test, modify and visualize a real life system without committing all the resources needed for the full project (Hardgrave, 1995: 116), to keep all project members on track (Barkin & Iansiti, 1994: 202 and Leonard-Barton, 1991: 65), to provide a vehicle by which different contributors to product development can articulate their concerns and issues, resolve questions quickly and unambiguously and reach agreements on the best ways to resolve conflicts and solve problems (Barkin & Iansiti, 1994: 202 and Wheelwright & Clark, 1992: 677), to narrow the range of feasible design parameters (Ulrich & Eppinger, 1995: 223), to anticipate future problems in the design, process, or satisfying customer needs (Wall, Ulrich & Flowers, 1991: 3) and to resolve crucial questions at every phase of the development cycle (Barkin & Iansiti, 1994: 227).

The growth in the use of prototypes to develop software has been significant during the last ten years. According to a survey done by Lingle *et al.* (1984), only 33% of the respondents used prototyping in the development of software in 1984. Necco, Gordon and Tsai (1987) found that this percentage had risen to 46% by 1987. In 1988, Carey and McLeod (1988) indicated 49% of the respondents to their survey on software projects used prototypes. Doke (1990) found that 61% of software engineers used prototypes in 1990.

Even though prototyping is increasingly used to develop software in industry, there is a lack of research on the relationship between software project outcomes and prototyping practice (Lichter *et al.*, 1994).

1.1 Objective

This thesis examines how dimensions of prototype management affect software project outcomes. Three software project outcomes are examined: project duration, schedule overrun and project manager's satisfaction that the software met customer requirements. The thesis aims to further the understanding of the nature and role of prototyping in industrial software development.

1.2 Prototype, Software Prototype and Prototype Process

In this section, the following terms are defined: prototype, software prototype and prototype process.

Prototype

A prototype is an easily modifiable and extensible working model of a proposed system, not necessarily representative of a complete system, which provides later users of the application system with a physical representation of key parts of the system before implementation (Boar, 1984).

Software Prototype

A software prototype is a dynamic model providing a communication tool for customer and developer that is far more effective than either narrative prose or static visual models for portraying functionality. Connel and Schafer (1989) define a software prototype to be i. functional after a minimal amount of effort, ii. a means of providing users of a proposed application with a physical representation of key parts of the system before system implementation, iii. flexible, modifications require minimal effort and iv. not necessarily representative of a complete system.

Prototype process

The prototype process is a series of iterations that occur during the development of a product, whereby each prototype embodies predicted end product specifications with greater scope and accuracy (Bischofberger & Pomberger, 1992; Budde, Kautz, Kuhlenkamp & Züllighoven, 1992; Carey & Currey 1989; Luqi & Royce, 1991; Maras, Riel, Nowlan, Menzel, & Mahoney, 1994 and Vonk, 1990). The prototype development process can be conceptualized as a life-cycle within a life-cycle (Bischofberger & Pomberger, 1992: 21).

The software prototyping process has the following characteristics: 1. prototypes

are constructed at an early stage in the development process, 2. relevant problems are clarified by experimentation and 3. prototypes provide a common basis for discussion between developers, users and other groups (Budde *et al.*, 1992: 9).

1.3 Contribution to Academic Literature

Most of the studies on prototyping have focused on student and experimental projects in academic settings (Blum, 1986 and Boehm, Gray & Seewaldt, 1984) and on case studies of software development projects (Lichter *et al.*, 1994). Little is known about how to manage prototyping and how the dimensions of prototype management affect outcomes of software development projects.

To my knowledge this is the first empirical study on how prototype management practice affects software project outcomes.

This study will add to the understanding of prototype management practice. The research focus is on the relationship between prototype management practice and the outcomes of software development projects.

1.4 Contribution to Management Practice

Traditionally, prototypes have been viewed by managers as technical tools. Managers, however, are under increasing pressure to produce complex software products with fewer design errors, in a short time and at a low cost. The results from this thesis will allow managers to further understand how prototypes can be used as management tools that can improve product development performance.

1.5 Organization of the Thesis

Chapter 2 provides the rationale for the hypotheses examined in this thesis. These hypotheses relate various aspects of prototype development to three outcomes of software development projects: project duration, schedule overrun and project manager's satisfaction that the software met customer requirements.

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

Chapter 4 presents the results of the research. This chapter is divided into nine sections. Section 4.1 provides the response rate for the questionnaire and the breakdown of the projects in the sample by geographical area. Section 4.2 identifies the tools and languages used to prototype. This section is necessary to understand the technical environment in which the data were gathered. Sections 4.3 to 4.5 present the descriptive statistics. Sections 4.4 and 4.5 also present the results of the ordinary least squares regression analyses and zero-order correlation analyses among dependent and control variables. Section 4.6 provides an exploratory data analysis. Sections 4.7 and 4.8 present a summary of results of the ordinary least squares regression analyses and zero-order correlation analyses required to examine the hypotheses. Section 4.9 presents a summary of significant results from the ordinary least squares regression analyses and zero-order correlation analyses. Section 4.10 presents the results of the stepwise regression analysis. Section 4.11 provides a list of important decisions made to shorten project duration, reduce schedule overrun and increase the project manager's satisfaction.

The results of the research and the limitations of the research approach are discussed in Chapter 5.

CHAPTER 2

LITERATURE REVIEW AND HYPOTHESES

Two bodies of literature were reviewed: software management and product development. Section 2.1 provides a classification of software prototypes. Section 2.2 examines the three project outcomes used as dependent variables in the thesis. Some of the hypotheses require that stages of the software development process be identified. Section 2.3 defines the stages that were used in the thesis. The hypotheses are developed in Section 2.4 and summarized in Section 2.5. The aspects of prototype management that were identified in the literature and not examined in this thesis are provided in Section 2.6.

2.1 Classification of Software Prototypes

Typically, there are two ways to classify software prototypes: by use of the prototype and by approach used to build the prototype. Software prototypes can be classified according to their use as: presentation prototypes, prototype proper, breadboards and pilot systems (Lichter *et al.*, 1994). These software prototypes can be used together. A presentation prototype is used to show the customer the feasibility of the future application system or demonstrate the functionality of the user interface. A prototype proper is used to illustrate specific aspects of a problem. A breadboard is developed primarily to address construction-related questions. The pilot system is the core of the application system.

There are three major approaches to the construction of software prototypes: evolutionary, throwaway and mixed.

Evolutionary Prototype

In the evolutionary approach to prototyping, many or most parts of the prototype are retained and the prototype eventually evolves to become the core of the end product. The goal of evolutionary prototyping is that of incremental system development (Bischofberger & Pomberger, 1992: 17).

Evolutionary prototypes allow the rapid clarification of requirements through frequent cyclical iterations. Continuous iterations are central to the evolutionary prototyping process. These iterations aid in solving software development problems throughout the process.

Luqi and Royce (1991) describe the evolutionary prototyping process as a series of prototypes that converge on an acceptable version of system behaviour via client feedback from prototype demonstrations (Luqi & Royce, 1991: 79). This allows the prototype team to rapidly narrow the scope of the system behaviour by creating prototypes that are close representations of the final software product. Evolutionary prototyping is a process to design, test and validate the system whereby all or part of the prototype is retained (Gordon & Bieman, 1995: 86).

Throwaway Prototype

In the throwaway approach to prototyping, the components used in the construction of a prototype are not built to be reused in subsequent prototype construction efforts. Throwaway prototypes are not designed to be part of the end-product's building blocks. They are expendable assets.

A throwaway prototype is a prototype that is discarded and not used in the delivered

product (Gordon & Bieman, 1995: 85-86); the code developed is thrown away (Luqi & Royce, 1991: 79). Throwaway prototypes are built "quick and dirty", used to clarify specifications in the early stages of the software development process and then they are thrown away (Davis, 1992: 71, Table 1).

Mixed Prototype

A mixed prototype is an operational combination of evolutionary and throwaway prototypes. The evolutionary prototype is the base of the product's architecture. Throwaway prototypes are built on top of the evolutionary prototype.

A mixture of evolutionary and throwaway prototypes can be used in the same software development project (Budde *et al.*, 1992 and Lichter *et al.*, 1994). Lichter *et al.* (1994) propose that finding the right mixture of prototypes improves the development process (Lichter *et al.*, 1994: 825). Combining throwaway and evolutionary prototypes into one operational prototype provides engineers benefits associated with both throwaway and evolutionary approaches (Bischofberger & Pomberger, 1992 and Davis, 1992). According to Davis (1992), throwaway prototypes work very well in isolation to verify relatively small parts of complex problems while evolutionary prototypes work well when most of the critical systems are well understood (Davis, 1992: 71). Therefore, Davis (1992) proposes the use of one operational mixed prototype. A mixed prototype is constructed by layering throwaway prototypes on top of an evolutionary prototype. The evolutionary prototype is constructed using requirements that are well understood. Throwaway prototypes are used to define and refine customer specified features until they are well understood by the development team (Davis, 1992: 74).

The practical implications of the mixed approach to prototyping are not yet well

understood. Moreover, the use of the mixed prototyping approach is time consuming.

2.2 Project Outcomes

Three measures of outcomes of software development projects will be used as dependent variables in this research: project duration, schedule overrun and project manager's satisfaction that the software met the customers requirements. The first two are frequently used as measures of success of development projects. The third is a measure of satisfaction with the project. This third measure is used in studies of group performance (Gladstein, 1984).

Griffen and Page (1993) grouped the success measures used in 75 studies of product development into five categories: customer acceptance, financial performance, product level, firm-based and program. Project duration and schedule overrun were two project outcomes measures included in the product level category. Since it was not possible to gain access to the respondents' customers, customer acceptance measures could not be used in this research. Measures from the other three categories were not used because they correspond to different levels of aggregation to that selected for this study. The level of aggregation used in this research is the project. Thus, using measures appropriate to the firm, program and the financial situation of the firm were excluded from this research.

Project duration

Project duration was identified as being critical to product development success and ultimately to the firm's survival (Eisenhardt & Tabrizi, 1995). Typically, the project's duration is measured as the interval marked by the start and end dates of the project. The

CHAPTER 2 • LITERATURE REVIEW AND HYPOTHESES

start date is set by the date at which the first meeting to consider the development of the product and to assign personnel to assess the feasibility of the product is held. The end date is the date at which the product is commercially released into the market place (Eisenhardt & Tabrizi, 1995: 100).

Schedule overruns

Gopalan (1995), Jenkins, Naumann and Wetherbe (1984), Paulish and Carleton (1994) and van Genuchten (1991) examined schedule overruns of software projects. Schedule overruns can be measured in months, weeks, cost or effort (in person hours). Typically, the magnitude of schedule overruns is calculated as the difference between actual project duration and estimated project duration (Gopalan, 1995: 72).

Project manager's satisfaction that the software met customer requirements

Studies of group effectiveness use satisfaction with the project as an output of group process (Gladstein, 1984: 502). Gladstein (1984) used three measures of satisfaction: satisfaction with the team, satisfaction with meeting customer needs and general satisfaction with extrinsic rewards and work. I have selected the second of these three variables as a dependent variable to be used for this research.

Gladstein (1984) had respondents rate their satisfaction with the project. Respondents rated how satisfied they were with dealing with the customer and meeting customer needs. Seven point Likert scales were used to measure the responses.

2.3 Stages of Software Development

The intent was to examine relationships that exist at different stages of software development. Thus, there was a need to define stages in an unambiguous manner. There is no agreement as to what are the appropriate stages for all software development processes. Bischofberger and Pomberger (1992) used six stages to define their software development process model, Boar (1984) and Boehm (1976) used seven stages, the United States of America's Department of Defense (1988) and Eisenhardt and Tabrizi (1995) used eight stages.

The stages used in this thesis were developed by combining the insights generated from the literature with those from four experienced software developers. The stages are shown in Figure 1 and are defined as follows:

- Stage one: begins with the date personnel are assigned to analyze requirements of the software to be produced and ends with the completion of basic software requirements
- Stage two: begins with the completion of basic software requirements and ends with the completion of the software's high level design
- Stage three: begins with the completion of the software's high level design and ends with the start of the field testing of the software
- Stage four: begins with the start of the field testing of the software and ends with the release of the software to users.

Figure 1
The Software Development Process

The stages in Figure 1 have been defined in terms of events rather than dominant development activities. These stages have not been named on purpose. The reason for this is that firms use different names to describe the stages that are used to develop software. It was difficult to name stages without generating ambiguity. The use of events divided the development process in an unambiguous way.

2.4 Testable Hypotheses

For reasons of brevity and clarity the words "software project success" will be used in lieu of the three dependent variables selected for this study. This will avoid making repetitive statements in the text. However, it should be clear that this is not a study of development project success factors.

Number of Prototypes

The number of prototypes built during development has been recognized in the literature as a key attribute of the prototyping process (Barkin & Iansiti, 1994; Clark & Fujimoto, 1991; Pisano, 1994 and Ulrich & Eppinger, 1995). The number of prototypes that need to be built depends on many factors such as the familiarity of the development team with the software application, the size and complexity of the intended system and the cost of each prototype (Barkin & Iansiti, 1994; Carey & Currey, 1989; Luqi & Royce, 1991; Pisano, 1994 and Ulrich & Eppinger, 1995).

Ulrich and Eppinger (1995) offer a benchmark on whether or not to add a prototype: If the required time for the prototype phase is less than the savings in duration of the subsequent activity, then this strategy is appropriate (Ulrich & Eppinger, 1995: 225).

The effectiveness of a project is strongly tied not only to the timing and frequency of prototypes, but to their quality as well (Barkin & Iansiti, 1994: 212). Barkin and Iansiti (1994), Clark and Fujimoto (1991) and Wheelwright and Clark (1992) suggest that producing many prototypes early will increase the feedback from all functions involved, reduce the number of errors detected in each iteration and shorten product project duration.

When multiple prototypes are developed in large projects, engineers are better able to adjust to new information and are less likely to become too attached to a particular design variation. Development teams that create multiple prototypes will have more confidence in their designs than those teams that do not. These teams will be able to settle on a design faster than teams that do not prototype. The increased learning from constructing multiple prototypes may translate into shorter project duration, smaller schedule overrun and greater project manager's satisfaction that the software met customer requirements. In formal

terms:

Hypothesis 1: The number of prototypes used throughout the development process is positively associated with software project success

Engineering involvement in software and prototype development

The integration of the prototyping team(s) and the software development team may lead to project success in the same way that the integration of design and manufacturing teams does. One of the ways to achieve this integration is to require engineers to be members of both the software development team and the prototype development team. This type of integration invariably increases each teams problem solving capabilities (Nadler & Tushman, 1992: 144). Engineers will be better able to integrate lessons learned in one development environment into the other in more stages of the process.

The individuals that are members of both software and prototype development efforts can become fluent in the language of each of the teams and can make sure that the translation and communication between teams is done effectively. The better the translation and communication between software developers and prototype developers the faster robust designs can be developed. Valuable time is not lost uncodifying and recodifying information from one group's language to that of another. For example, information passed from prototype developers to software developers can be codified in the language with which software developers are familiar.

If the same individuals are involved in many steps of prototype development as well as many steps of software product development, they are more likely to identify and solve future problems sooner in the process. Thus, the larger the number of software development engineers who are involved in prototyping the shorter the project duration, the

smaller the schedule overrun and the greater the project manager's satisfaction that the software met customer requirements. In formal terms:

Hypothesis 2: The extent to which engineers are involved in both software development and prototype development is positively associated with software project success

External involvement

External involvement corresponds to the participation of employees of customers, suppliers and senior managers in the development and assessment of prototypes throughout the development project. The involvement of members of groups external to the software project in the development and assessment of prototypes is expected to lead to shorter duration, smaller schedule overrun and greater project manager's satisfaction with the product.

Customers, suppliers and senior managers are stakeholders external to the project team. They are motivated to contribute to the success of the project through the construction of good designs. These external stakeholders should be involved as early as possible in the prototyping process (Barkin & Iansiti, 1994). The primary objective of external involvement is to detail and validate the product's specifications while the cost of changes in specifications is low (Budde *et al.*, 1992; Lichter *et al.*, 1994; Ulrich & Eppinger, 1995; Vonk, 1990 and Wheelwright & Clark, 1992).

Opinions about the effect of the involvement of external groups in the prototyping process on project outcomes vary. Gordon and Bieman (1995) found that user involvement in the requirements definition and testing of prototypes has a positive effect on the product by increasing the likelihood that user needs will be met (Gordon & Bieman,

1995: 92). Vonk (1990) indicates that allowing customers to assess a prototype strongly influences the success of the prototyping process (Vonk, 1990: 90).

Carey and Currey (1989) argue that the involvement of users is an important element in the development of software prototypes (Carey & Currey, 1989: 30). Software companies that prototype do so to gain the benefit of user involvement and to help users clarify requirements (Carey & Currey, 1989: 30). Lichter *et al.* (1994) found that the involvement of end-users is of major importance for the success or failure of the prototyping project (Lichter *et al.*, 1994: 830). Most advocates of prototyping point to user involvement as one of the major benefits provided by prototypes (Clark & Fujimoto, 1991; Lichter *et al.*, 1994 and Wheelwright & Clark, 1992). Some argue that customer involvement allows the development team to record customer perceptions and feedback quickly (Barkin & Iansiti, 1994; Gordon & Bieman, 1995 and Wheelwright & Clark, 1992). This decreases project duration.

However, others argue that customer involvement in prototype development may not be required. To Budde *et al.* (1992), the use of prototypes does not necessarily mean that the customer should be involved in testing the prototype. The in-house prototype is built and prototyped internally by the manufacturer. Designers try to anticipate user needs until they think the prototype is ready for field testing.

Two major problems with the involvement of customers in the assessment of field prototypes have been identified in the literature. First, customers may want to keep the prototype and assume that it is a good substitute for the end-product (Bischofberger & Pomberger, 1992; Budde *et al.*, 1992; Carey & Currey, 1989; Davis, 1992 and Hardgrave, 1995). Second, as users work with the field prototypes they detect opportunities for new features and demand that the scope of the development project change (Davis, 1992: 72).

Feature creep is a challenge when managing the involvement of users in the construction and assessment of prototypes (Carry & Currey, 1989: 30).

Barkin and Iansiti (1994) found that supplier involvement must take place early in the design and development of the prototype. This allows the supplier to provide input and plan for the construction of quality parts. Eisenhardt and Tabrizi (1995) found that supplier involvement reduces project duration. If suppliers are involved in many steps of the process, then they are more likely to detect future problems, such as specification errors, early in the process when they are easier to fix (Eisenhardt & Tabrizi, 1995: 89). There are many changes that take place during the design specification stage. Involving the suppliers in the development of prototypes allows them to keep up to date with the latest product specifications. Since suppliers have specialized knowledge they are more likely to catch design errors unique to their domain early in the development process (Barkin & Iansiti, 1994; Clark & Fujimoto, 1991 and Wheelwright & Clark, 1992) and contribute to project success.

Senior management involvement in the prototyping process has been cited in the literature as being required for the implementation and support of prototype development (Vonk, 1990). However, some authors indicate that managerial involvement in the prototyping process can extend the duration of the prototyping process.

Typically, senior managers review the business case for the investment in prototyping at the beginning of the process and support prototype development from a distance. Moreover, senior managers tend to view prototypes as technical tools for engineers rather than management tools (Clark & Fujimoto, 1991). The pressure to produce products quicker and to satisfy ever increasing customer requirements is forcing senior managers to use prototypes as management tools. The more managers use

prototypes as management tools the more they are required to be involved in the prototype development process. Further, senior managers who are involved in prototype development gain a better understanding of the complete prototyping process.

Davis (1992) found that managers usually insist that they know what the customer needs are. This may lead to significant rework later on. According to Davis (1992), engineers responsible for prototyping can be successful only if they are free to satisfy the customer needs on site experimentation, without management intervention (Davis, 1992: 78). In formal terms:

Hypothesis 3: External involvement in prototype development is positively associated with software project success

Functional involvement

The involvement of personnel from the company's functions in prototype development has been identified as an important factor for the successful development of the product (Barkin & Iansiti, 1994; Gupta & Wilemon, 1990 and Wheelwright & Clark, 1992). Functional involvement allows the prototype to be developed incorporating a variety of specialized inputs (Wheelwright & Clark, 1992). For example, marketing supplies customer input, design provides input on the software's structure and engineering can provide input on the product's functionality.

Functional involvement may also allow the prototyping process to be more productive since downstream activities such as marketing are involved early in the prototyping process.

The use of specialized inputs from such functions as design, marketing, sales and purchasing should accelerate prototype development. Prototypes incorporate the concerns of the various functions and thus become more representative of the end product in a

shorter amount of time (Barkin & Iansiti, 1994; Gupta & Wilemon, 1990 and Wheelwright & Clark, 1992). Prototypes are used by the development team to map out an uncertain domain. Each function is able to focus on specific aspects of the domain akin to their background. Therefore, the gathering and assimilation of information using input from the functions will allow team members to have a shared view which is concrete and relevant. This should accelerate product development, reduce schedule overrun and increase the project manager's satisfaction that the software met customer requirements.

Opinions about involvement of marketing and sales personnel in the construction and assessment of prototypes vary. The involvement of marketing and sales personnel in the prototype development process has been found to be desirable by Gupta and Wilemon (1988), Hise, O'Neal, Parasuraman and McNeal (1990) and Souder (1988). Barkin and Iansiti (1994), however, found that when marketing and sales are involved late in the prototype development process the product was over budget and encountered serious delays. The involvement of marketing and sales too late in the development of the prototype may have adverse consequences for the firm. At the same time, the researchers found that in technology driven companies where engineers dominate the development team, input from marketing and other less technical sources was often disregarded or devalued (Barkin & Iansiti, 1994: 236).

Researchers in software development have reported that there are concerns about the involvement of marketing and sales personnel in the process used to develop in-house prototypes (Gordon & Bieman, 1995 and Lichter *et al.*, 1994). Marketing and sales may add little information of value when the software development team knows what the customer needs are. The involvement of marketing and sales may result in increased prototype project duration, because these sources usually supply their own view of users'

needs which may not be aligned with the real needs that users have (Lichter *et al.*, 1994: 830).

Another reason for limiting the involvement of marketing and sales personnel in the in-house prototyping process is that they may pass along inappropriate expectations to customers after seeing working prototypes (Gordon & Bieman, 1995: 93). In formal terms:

Hypothesis 4: Functional involvement in prototype development is positively associated with software project success

Number of customer sites used in field testing

Field prototypes are working representations of the product which embody most, if not all of the product's ultimate functionality. Field prototypes are used to test the prototype's features, functionality and robustness under real-time conditions. Integrating the prototype into the customer's environment allows the customer and the prototype team to identify and further refine product specifications and functionality requirements (Barkin & Iansiti, 1994; Budde *et al.*, 1992; Ulrich & Eppinger, 1995; Vonk, 1990 and Wheelwright & Clark, 1992). A large number of customer sites contributing to the assessment of field prototypes may contribute to project success. In formal terms:

Hypothesis 5: The number of customer sites involved in the assessment of prototypes is positively associated with software project success

Prototyping effort

More effort spent prototyping should increase the likelihood of project success. The prototype serves as a focal point of the product development process where by all the

prototype team members can communicate and understand one another (Leonard-Barton, 1991; Ulrich & Eppinger, 1995 and Vonk, 1990). Prototypes enhance the level of communication among the prototype team members (Barkin & Iansiti, 1994; Bischofberger & Pomberger, 1992; Budde *et al.*, 1992; Clark & Fujimoto, 1991; Leonard-Barton, 1991; Lichter *et al.*, 1994; Luqi & Royce, 1991; Ulrich & Eppinger, 1995; Vonk, 1990 and Wheelwright & Clark, 1992). The use of prototypes allows the software development team to better understand the problem domain (Budde *et al.* 1992; Carey and Currey 1989; Gordon and Bieman, 1995; Luqi and Royce, 1991 and Vonk, 1990) and to identify design problems (Barkin & Iansiti, 1994).

Effort to develop and assess prototypes can be measured in two ways: person-months or dollar cost. Person-months is used to measure effort in software development estimation models (Boehm, 1981) and in software scheduling models. Budget figures incorporate the wages paid by the development organization. The budget measure, unlike the person-month measure, reflects wage differentials paid by different organizations. In formal terms:

Hypothesis 6a: The amount of effort dedicated to prototyping throughout the development process is positively associated with software project success

The use of prototypes during the early stages of the development process provides earlier learning experiences that drive the project more rapidly to convergence and ultimately to project success (Barkin & Iansiti, 1994: 206). Prototypes constructed prior to the completion of the software specification and the start of detailed design and coding can be used by the development team to detect engineering problems. Early detection of problems is much cheaper than detecting errors towards the end of the development cycle

(Barkin & Iansiti, 1994; Bischofberger & Pomberger, 1992; Clark & Fujimoto, 1991; Gordon & Bieman, 1995 and Maras *et al.*, 1992).

Early prototyping requires a larger investment up front to reduce longer-term risk (Barkin & Iansiti, 1994: 206). The construction of prototypes at the product definition stage (Barkin & Iansiti, 1994; Clark & Fujimoto, 1991 and Wheelwright & Clark, 1992) or the requirements definition stage (Maras *et al.*, 1994 and Vonk, 1990) is considered key to product success. Early prototypes allow the team to quickly define and narrow the scope of the system requirements. Thus, investment in prototyping, particularly during the early phases of software development, should result in software project success. In formal terms we state the following three hypotheses which are related to Hypothesis 6a.

- Hypothesis 6b:** The amount of effort dedicated to prototyping prior to the completion of basic software requirements (stage 1) is positively associated with software project success
- Hypothesis 6c:** The amount of effort dedicated to prototyping during the stage that starts with the completion of basic software requirements and ends with the completion of high level design (stage 2) is positively associated with software project success
- Hypothesis 6d:** The amount of effort dedicated to prototyping during the stage that starts with completion of high level design and ends with the start of the field testing of the software (stage 3) is positively associated with software project success

Extent to which prototypes embody the product

Leonard-Barton (1991) classifies prototypes into five groups according to how well they embody the final product: 2-D (flat models); Non-Functional 3-D Models; Functional Prototypes; User Test Models and Organization / System Models. As the prototyping process progresses, each prototype calls upon a growing number of senses and the

language used to describe the prototype becomes more removed from the language of the specialized experts that created it.

How well each prototype embodies the final product at each stage of development is an attribute that may give an indication of the prototyping strategy used by the project manager. Lichter *et al.* (1994) propose that the prototype's architecture must match that of the final application system early in the development process. Budde *et al.* (1992) found that when using an evolutionary system development approach to software development, the prototypes are completed as much as possible using operational software versions. This allows and facilitates evaluation of the models early in the development process (Budde *et al.*, 1992: 51).

As the software prototyping process progresses, the prototype becomes more reliable and secure, performs better and operates under hardware and software conditions that better resemble field operating conditions (Vonk, 1990).

A prototype embodies a portion of the functionality of the end product. The more the prototype embodies the final product the easier it is to detect costly design errors early in the development process (Bischofberger & Pomberger, 1992; Barkin & Iansiti, 1994; Clark & Fujimoto, 1991; Gordon & Bieman, 1995; Hardgrave, 1995; Leonard-Barton, 1991 and Maras *et al.*, 1992). The more the prototype embodies the end product's functionality, quality and performance the more robust tests can be performed earlier in the development process (Barkin & Iansiti, 1994; Gordon & Bieman, 1995; Hardgrave, 1995 and Maras *et al.*, 1992). This allows the development team to integrate various aspects of the design earlier in the development process. In formal terms:

Hypothesis 7a: The extent of the system embodied in the prototypes at the completion of basic software requirements (stage 1) is positively associated with software project success

The following two hypotheses are related to Hypothesis 7a. They focus on the extent of the system embodied in the prototypes at different stages of the development cycle.

Hypothesis 7b: The extent of the system embodied in the prototypes at the completion of high level design (stage 2) is positively associated with software project success

Hypothesis 7c: The extent of the system embodied in the prototypes at the start of the field testing of the software (stage 3) is positively associated with software project success

2.5 Summary of Hypotheses

The hypotheses to be examined in this thesis are:

Hypothesis 1: The number of prototypes used throughout the development process is positively associated with software project success

Hypothesis 2: The extent to which engineers are involved in both software development and prototype development is positively associated with software project success

Hypothesis 3: External involvement in prototype development is positively associated with software project success

Hypothesis 4: Functional involvement in prototype development is positively associated with software project success

Hypothesis 5: The number of customer sites involved in the assessment of prototypes is positively associated with software project success

Hypothesis 6a: The amount of effort dedicated to prototyping throughout the development process is positively associated with software project success

Hypothesis 6b: The amount of effort dedicated to prototyping prior to the completion of basic software requirements (stage 1) is positively associated with software project success

- Hypothesis 6c:** The amount of effort dedicated to prototyping during the stage that starts with the completion of basic software requirements and ends with the completion of high level design (stage 2) is positively associated with software project success
- Hypothesis 6d:** The amount of effort dedicated to prototyping during the stage that starts with completion of high level design and ends with the start of the field testing of the software (stage 3) is positively associated with software project success
- Hypothesis 7a:** The extent of the system embodied in the prototypes at the completion of basic software requirements (stage 1) is positively associated with software project success
- Hypothesis 7b:** The extent of the system embodied in the prototypes at the completion of high level design (stage 2) is positively associated with software project success
- Hypothesis 7c:** The extent of the system embodied in the prototypes at the start of the field testing of the software (stage 3) is associated with software project success

2.6 Aspects of Prototype Management Identified in the Literature that Were Not Examined in this Thesis

Not all of the aspects of prototype management identified in the literature were examined in this thesis. Those aspects of prototype management that were identified in the literature as being related to project outcome that were not examined in the thesis are listed in Table 1.

Table 1
Aspects of Prototype Management Identified in the Software and Product Development Literature Not Examined in the Thesis

Dimension	References
Prototype project duration	Barkin and Iansiti (1994) Carey and Currey (1989) Maras <i>et al.</i> (1994) Wheelwright and Clark (1992)
Prototype-related milestones	Ulrich and Eppinger (1995)
Project leader's understanding of prototyping	Gordon and Bieman (1995) Vonk (1990)
Development team's commitment to prototyping	Barkin and Iansiti (1994) Budde <i>et al.</i> (1992) Clark and Fujimoto (1991) Cummings and Worley (1993) Vonk (1990) Wheelwright and Clark (1992)

CHAPTER 2 • LITERATURE REVIEW AND HYPOTHESES

Level of communication	<p>Barkin and Iansiti (1994)</p> <p>Bischofberger and Pomberger (1992)</p> <p>Budde <i>et al.</i> (1992)</p> <p>Leonard-Barton (1991)</p> <p>Lichter <i>et al.</i> (1995)</p> <p>Luqi and Royce (1991)</p> <p>Ulrich and Eppinger (1995)</p> <p>Wheelwright and Clark (1992)</p> <p>Vonk (1990)</p>
Customer experience with prototypes	<p>Budde <i>et al.</i> (1992)</p> <p>Hardgrave (1995)</p> <p>Vonk (1990)</p>
Prototype development overlap	<p>Clark and Fujimoto (1991)</p> <p>Ulrich and Eppinger (1995)</p>
Planning	<p>Budde <i>et al.</i> (1992)</p> <p>Clark and Fujimoto (1991)</p> <p>Lichter <i>et al.</i> (1994)</p> <p>Ulrich and Eppinger (1995)</p> <p>Vonk (1990)</p>
Power of the project leader	<p>Clark and Fujimoto (1991)</p> <p>McDonough (1993)</p> <p>Wheelwright and Clark (1992)</p>

CHAPTER 2 • LITERATURE REVIEW AND HYPOTHESES

The researcher was unable to operationalize prototype-related milestones, prototype development overlap and development team's commitment to prototyping. The researcher was informed that he could not have access to customers of the firms that provided data for the study. Thus, customer experience with prototyping could not be examined. In addition, customer opinions of project success could not be used as a dependent variable.

The researcher found that project managers do not keep data on the duration of the planning prototypes stage.

The researcher did not examine power of the project leader and project leader's understanding of prototyping because of response bias. The researcher concluded that the data on these dimensions provided by project leaders would be biased and lack validity.

The researcher concluded that to measure level of communication throughout the software development cycle required a longitudinal study and resources that were outside of his financial capability.

CHAPTER 3

RESEARCH DESIGN

3.1 Unit of Analysis

The unit of analysis is a development project that used one or more prototypes to develop a software product. A software product can be a custom system, a semi-custom system, or a standard product.

3.2 Sample

The sample is comprised of 40 software development projects drawn from companies that develop software in Canada. The projects in the sample were completed within the three years prior to October 1, 1995.

3.3 Company, Project and Respondent Selection

The researcher sent the letters shown in Appendix A to the presidents or relevant vice presidents of the companies involved in software development listed in the 1993 Industry and Science Canada Completed Profiles of Canada's Premier Software Companies and the 1995 Ottawa-Carleton Economic Development Corporation directory asking them to participate in the study. The letter requested their cooperation in the collection of data for this research. The researcher then followed up with telephone calls.

The researcher asked company executives to select up to three recently completed software development projects. The researcher then randomly selected the project that was to be part of the sample. Thus, only one development project from each company that agreed to participate was included in the sample.

The project leaders of the projects selected were sent the letter shown in Appendix B. In this letter the researcher requested that the project leaders provide data for the project selected. These project leaders provided the answers to the questionnaire included as Appendix C. The six page questionnaire had instructions and four sections. Section 1 requested general information of the project. Section 2 focused on questions that pertained to prototype development. Section 3 asked about the tool and languages used in the technical environment. This section was necessary to characterize the context in which the data were being gathered. Section 4 included three open ended questions. These questions addressed prototype management decisions linked to shortening project duration, reducing schedule overrun and increasing the project manager's satisfaction that the software met customer's requirements.

3.4 Time of Data Collection

Data were collected from October 1995 to February 1996. Data were collected using the questionnaire shown in Appendix C.

3.5 Data Collection Plan

3.5.1 Piloting the Questionnaire

The questionnaire was tested with the assistance of four managers of software development projects. They were asked to provide their comments on the questionnaire. The four initial respondents rephrased the questions aloud in their own words. Then the researcher asked them to interpret key words used in the questionnaire. Their comments were recorded and after discussion with members of the thesis committee changes were

incorporated into the questionnaire. The individuals that participated in the piloting of the questionnaire were not part of the sample respondents.

3.5.2 Data Collection

After securing the cooperation of company presidents or vice presidents, the researcher sent the questionnaire in Appendix C to the project leaders of the software development projects. The cover letter provided an overview of the research, assurances of the confidentiality of the responses and the names of the researcher and the thesis supervisor whom the software project leader could contact with questions or concerns at any time. The questionnaire provided explicit instructions and defined key terms in order to facilitate accurate completion by the respondent.

After the questionnaires were received, the researcher reviewed them for problems and completeness.

3.6 Measures

Project duration

Project duration was measured as the number of weeks between the start and the end of the software development project. The start of the software development project is the date at which personnel were assigned to analyze the requirements for the software. The end of the software development project is the date at which the software was delivered to users.

Schedule Overrun

Schedule overrun was calculated as the difference in weeks between the date the software was delivered to users and the date that was estimated at the start of the project for the delivery of the software to users.

Project manager's satisfaction that the software met customer requirements

Respondents scores on 5 point Likert scales to the following question: "How satisfied were you that the software met customer requirements" were used to measure project manager's satisfaction with the product. The anchors of the scale were 1=very dissatisfied and 5=very satisfied.

Development Stages

The four stages used in the thesis are shown in Figure 1. The stages are as follows:

- Stage one: begins with the start of the project (date at which personnel were assigned to analyze the requirements for the software) to be produced and ends with the completion of basic software requirements
- Stage two: begins with the completion of basic software requirements and ends with the completion of high level design
- Stage three: begins with the completion of high level design and ends with the start of the field testing of the software
- Stage four: begins with the start of the field testing of the software and ends with the release of the software to the user community.

Project size

Project size was measured in three ways: number of lines of source code (new and reused) in the software, total cost of the development project and total number of person-months dedicated to the development project. Project size was used as the control variable in the regression analyses performed on the data.

Number of prototypes

Number of prototypes was measured directly by asking how many prototypes were constructed during the project. The number of prototypes includes the sum of all throwaway, evolutionary, or mixed prototypes used in the project.

Engineers and computer scientists that were members of both software and prototype development teams

The extent to which product development engineers were involved in the prototype process was measured directly by asking how many engineers and computer scientists were members of both the software team and the prototype development teams, full time and part time. Each engineer or computer scientist working part time was deemed to be equivalent to half an engineer or computer scientist working full time.

External involvement

External involvement corresponds to the participation of employees of customers and suppliers as well as senior managers in the development and assessment of prototypes. The involvement of customers, suppliers and senior managers during prototype development was measured as the number of stages in which members of these external

groups actually participated in the development and assessment of the prototypes. Involvement was defined as having one or more employees of the external group actively participating in meetings and design activities that affected prototype development. For each stage, external involvement was measured as the sum of the involvement of the external groups. External involvement for the project was measured as the sum of the external involvement of each of the four stages. The measure of external involvement used here is identical to the measure of supplier involvement used by Eisenhardt and Tabrizi (1995: 98).

Functional involvement

Functional involvement was measured by whether personnel from specific functions participated in the development and assessment of the prototype during the four stages of software development shown in Figure 1. The functions are 1. marketing / sales; 2. product management; 3. software design; 4. customer services; 5. quality and 6. software maintenance. For each stage of software development, the total number of participating functions were calculated. The functional involvement measure was calculated by adding the total number of functions in every stage. Involvement is defined as having one or more members of the function actively participating in meetings and design activities that affected prototype development. Functional involvement for the project was measured as the sum of the functional involvement of each of the four stages. The measure of functional involvement used here is identical to Eisenhardt and Tabrizi's measure of multifunctional teams (1995: 99).

Customer sites involved in field prototypes

The number of customer sites involved in the assessment of field prototypes was measured directly by asking how many sites were involved in the field testing of each prototype during the systems testing and operations stage.

Effort dedicated to prototyping

Effort dedicated to prototyping throughout the development process was directly measured by asking the respondents how many person-months were dedicated to the development and assessment of prototypes.

Respondents were asked to provide the fractions of the total person-months invested in the development and assessment of prototypes during each of the four stages. Effort dedicated to prototyping during a stage was calculated by multiplying the person-months dedicated to the development and assessment of the prototypes throughout the development process during by the proportions of the total person-months dedicated to prototyping.

System embodied in the prototypes

For each of the four stages of development, the extent to which the prototypes built embodied the target system was estimated as a percentage on an open-ended scale. This measure was a composite of the extent to which the prototypes built embodied the functionality, real-time performance, ease of use, reliability and quality of the end product. Each was estimated as a percentage on an open-ended scale.

3.7 Data Analysis

The 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. For example project duration was calculated as the difference between the date of the initial release of the software and the date personnel was assigned to analyze software requirements.
- Step 2.** Descriptive summary statistics were calculated for all of the variables, including the means, medians, standard deviations and correlations among the variables.
- Step 3.** Exploratory data analysis was performed using Pearson's zero-order correlation analysis and forward selection regressions. Pearson's zero-order correlation analysis was used to examine the associations between the independent variables. The independent variables were regressed against project duration and schedule overrun using project size as the control variable.
- Step 4.** For each hypothesis the independent variables were individually regressed against project duration and schedule overrun using project size as the control variable. Pearson's zero-order correlation analysis was used to examine the associations between project manager's satisfaction that the software met customer requirements 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. Stepwise regression was performed to develop the optimum predictor models for project duration and schedule overrun. The independent variables were regressed against project duration and schedule overrun 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. Outliers were identified and removed.

CHAPTER 4

RESULTS

Section 4.1 provides the response rate for the questionnaire and the breakdown of the projects in the sample by geographical area. Section 4.2 identifies the tools and languages used to prototype in the projects in the sample. Sections 4.3, 4.4 and 4.5 present the range, means, modes and standard deviations for the independent, dependent and control variables. Sections 4.4 and 4.5 also provide the zero-order correlations among dependent and control variables respectively. Section 4.6 provides an exploratory data analysis using zero-order correlation analysis and forward selection regression analysis. The zero-order correlation analysis was used to examine the associations between the independent variables. Sections 4.7 and 4.8 present the results of the ordinary least squares regression analyses and zero-order correlation analyses required to examine the hypotheses. Section 4.9 presents a summary of significant results from the ordinary least squares regression and zero-order correlation analyses. Section 4.10 presents the results of the stepwise regression analysis. Section 4.11 provides the list of important decisions made to shorten project duration, reduce schedule overrun and increase the project manager's satisfaction that the software met customer requirements. The significance level sought was $p < .05$.

4.1 Response Rate and Breakdown

Presidents or vice presidents of 107 companies that build prototypes as part of their software development process were asked to fill-out the questionnaire. Questionnaires were sent to 60 project leaders. Forty usable questionnaires were returned by project

leaders by either fax or mail. Thus, the response rate for the questionnaire was 37 percent. The most frequent reason given for not returning the questionnaire was lack of time to gather the data requested.

The breakdown of the 40 projects by geographical area was: 67.5 % from Ottawa-Carleton, 17.5 % from Toronto, and 15.0% from the rest of Canada. There is no claim that the sample is representative of Canadian software companies.

4.2 Tools and Languages Used to Prototype

This section is necessary to understand the context in which the data were gathered.

Table 2a provides a breakdown of the tools used to develop prototypes in the sample projects. These tools can be classified as being general purpose or specialized. They can also be classified on the basis of whether or not they implement the object-oriented approach to software development.

Table 2a
Tools Used to Prototype

Tool used	Object-oriented	Non object-oriented	Both	Did not know	Number of observations
<u>General Purpose</u>					
Operating System / Environment / CASE Tool	2	8	5	1	16
Programming	10	5	2	0	17
<u>Specialized</u>					
Graphical user interface	1	7	0	0	8
Database	0	10	0	0	10
Spreadsheet	0	1	0	0	1
Other	2	0	0	7	9
Overall Total	15	31	7	8	61

Table 2a shows that 33 of 61 of the tools used to develop prototypes (54%) were of the general purpose type. Of the 33 general purpose tools, 12 incorporated the object-oriented paradigm, 13 did not. Seven tools incorporated the object oriented paradigm and at least one non-object oriented approach to software development. Of the 28 specialized tools only 3 incorporated the object-oriented approach.

Table 2b provides a breakdown of the type of language used to prototype. Of the 33 languages used to develop prototypes, 26 were programming languages. Of the 26 programming languages 15 (58%) were object-oriented (Visual C++ or C++) and 10 (38%) were non object-oriented (Visual C or C).

Table 2b
Languages Used to Prototype

Language used	Object-oriented	Non object-oriented	Did not know	Number of observations
Programming	15	10	1	26
SpreadSheet	0	1	0	1
Database	0	1	0	1
Graphical user interface	0	2	0	2
Operating System / Environment / CASE Tool	0	1	0	1
Other	0	0	2	2
Total	15	15	3	33

4.3 Independent Variables

Table 3 presents the range, means, modes, standard deviations and number of observations for the independent variables used in the study. Table 3 excludes the control variables used. Table 3 shows that the number of prototypes built in the software development projects in the sample ranged from 1 to 10. Fifty percent of the projects in the sample built only 1 prototype.

The number of engineers who were involved in both software development and prototype development during the project ranged from 0 to 15. Twenty-two percent of the projects had the equivalent of 1.5 engineers involved in both software development and prototype development.

External involvement throughout the development process ranged from 0 to 12. Ten percent of the sample projects did not have external involvement throughout the prototyping

CHAPTER 4 • RESULTS

process.

External involvement prior to the completion of basic software requirements (stage 1) ranged from 0 to 3. Fifteen of the sample projects had two external groups involved prior to the completion of basic software requirements (stage 1).

External involvement during stage 2 (starts with the completion of basic software requirements and ends with the completion of high level design) ranged from 0 to 3. Fifteen of the sample projects involved only one external group during stage 2.

External involvement during stage 3 (which starts with the completion of high level design and ends with the start of the field testing of the software) ranged from 0 to 3. Fifteen of the sample projects involved two groups during stage 3.

Functional involvement throughout the development process ranged from 0 to 24. Fifteen percent of the sample projects involved only four functions throughout the development process.

Functional involvement prior to the completion of basic software requirements (stage 1) ranged from 0 to 6. Thirty percent of the sample projects had two functions involved prior to the completion of basic software requirements (stage 1).

Functional involvement during stage 2 (starts with the completion of basic software requirements and ends with the completion of high level design) ranged from 0 to 6. Seventeen of the sample projects involved two functions during stage 2.

Functional involvement during stage 3 (which starts with the completion of high level design and ends with the start of the field testing of the software) ranged from 0 to 6. Nine of the sample projects involved two functions during stage 3.

The number of customer sites involved in field prototypes ranges from 0 to 20. Approximately one third of the projects in the sample did not involve customer sites in

CHAPTER 4 • RESULTS

prototype assessment.

The effort dedicated to prototyping throughout the development process ranged from .50 to 350 person-months. In five of the 40 sample projects only one person-month was allocated to prototyping.

The effort dedicated to prototyping prior to the completion of basic software requirements (stage 1) ranged from 0 to 111 person-months. No prototypes were built prior to the completion of basic software requirements in 11 of the sample projects.

The effort dedicated to prototyping during stage 2 (starts with the completion of basic software requirements and ends with the completion of high level design) ranged from 0 to 105 person-months. Seven of the sample projects did not prototype during stage 2.

The effort dedicated to prototyping during stage 3 (which starts with the completion of high level design and ends with the start of the field testing of the software) ranged from 0 to 141 person-months. Ten percent of the sample projects did not prototype during stage 3.

The extent of the system embodied in prototypes at the end of stage 1 ranged from 0 to 99%. Fifteen percent of the sample projects embodied 0% of the system in the prototype at the completion of basic software requirements.

The extent of the system embodied in prototypes at the end of stage 2 ranged from 0 to 100%. Fifteen percent of the sample projects embodied 0% of the system in the prototype at the completion of high level design.

The extent of the system embodied in prototypes at the start of stage 3 ranged from 0 to 100%. In 10% of the sample projects, the prototypes at the start of field testing of the software embodied 0% of the system.

Table 3
Range, Means, Modes, Standard Deviations and Number of Observations
for Independent Variables

Independent Variable	Range	Mean	Mode	S.D.	Number of Observations
Number of prototypes used throughout the development process	1-10	2.46	1.00	2.47	40
Number of engineers and computer scientists involved in both software development and prototype development	0-15	2.98	1.50	3.49	40
External involvement in prototype development					
• across all stages	0-12	5.65	8.00	3.26	40
• during stage 1	0-3	1.63	2.00	.98	40
• during stage 2	0-3	1.50	1.00	.96	40
• during stage 3	0-3	1.25	2.00	.95	40
Functional involvement in prototype development					
• across all stages	0-24	9.98	4.00	6.02	40
• during stage 1	0-6	2.68	2.00	1.61	40
• during stage 2	0-6	2.60	2.00	1.69	40
• during stage 3	0-6	2.43	2.00	1.85	40
Number of customer sites involved in the assessment of prototypes	0-20	2.80	0.00	4.18	40
Effort dedicated to prototyping					
• across all stages (person-months)	.5-350	34.54	1.00	76.18	40
• during stage 1 (person-months)	0-111	7.72	0.00	21.49	39
• during stage 2 (person-months)	0-105	8.17	0.00	21.77	39
• during stage 3 (person-months)	0-141	12.91	0.00	27.92	39
Extent to which the prototypes embodied the system					
• at the end of stage 1	0-99	29.38	0.00	28.53	40
• at the end of stage 2	0-100	57.39	90.00	33.09	40
• at the start of stage 3	0-100	79.47	100.00	33.19	40

4.4 Dependent Variables

Table 4 presents the range, means, modes, standard deviations and number of observations for the variables used to measure software project outcome.

The duration of each project in the sample ranged from 9.1 to 156 weeks. The schedule overrun encountered in the sample ranged from -2.00 to 124.4 weeks. Two of the sample projects were able to complete the project ahead of schedule. Seven of the 40 projects in the sample had no schedule overrun.

The project manager's satisfaction that the software met customer requirements ranged from 1 to 5.

Table 4
Range, Means, Modes, Standard Deviations and Number of Observations for Dependent Variables (Observations = 40)

Dependent Variable	Range	Mean	Mode	S.D.
Project duration (weeks)	9.1-156	62.4	11.00	42.7
Schedule overrun (weeks)	-2.0-124.4	24.9	0.00	31.1
Project manager's satisfaction that the software met customer requirements	1-5	3.9	4.00	1.1

Table 5 shows the zero order correlations among the three dependent variables.

Table 5
Correlations for Project Duration, Schedule Overrun and Project Manager's Satisfaction
that the Software Met Customer Requirements
(Observations = 40)

Dependent variable	Project duration	Schedule overrun	Project manager's satisfaction
Project duration (weeks)	1.00	.62***	.08
Schedule overrun (weeks)		1.00	.19
Project manager's satisfaction that the software met customer requirements			1.00

p<.05; *p<.01

4.5 Project Size

Table 6 presents the range, means, modes, standard deviations and number of observations for the three variables used to measure project size.

The total cost of each project in the sample ranged from \$8,750 to \$3,100,000. The total source lines of code ranged from 730 to 1,000,000. The total person-months dedicated to the project ranged from 2 to 350.

Table 6
Range, Means, Modes, Standard Deviations and Number of Observations for the Variables
Used to Measure Project Size

Control Variable	Range	Mean	Mode	S.D.	Number of observations
Total cost of the project	\$8,750-3,100,000	522,667	300,000	829,365	40
Total source lines of code	730-1,000,000	178,001	5,000	245,050	37
Total person-months dedicated to the project	2-350	64	2	91	40

Table 7 shows the zero-order correlations among the three control variables. Calculations were performed with 40 observations for total cost and total person-months and with 37 observations for total code. The results in Table 7 show that total person-months is strongly associated with total cost. For this reason one of the control variables, total cost was excluded from the regression analysis.

Table 7
Correlations for Total Code, Total Cost and Total person-months
(Observations = 40)

Variable	Total code	Total cost	Total person months
Total code	1.00	.07	.29
Total cost		1.00	.66***
Total person-months			1.00

p<.05; *p<.01

4.6 Exploratory Data Analysis

The zero order correlation matrix for the independent variables is shown in Appendix D. Calculations were performed with 40 observations for all the independent variables except for effort dedicated to prototyping during stages 1, 2 and 3, where 39 observations were used. The results indicate that all external involvement variables are strongly associated with one another. Functional involvement across all stages is positively associated at $p<.01$ with external involvement during stage 1. Functional involvement during stage 1 is positively associated at $p<.01$ with external involvement during stage 1. Functional involvement during stage 3 is positively associated at $p<.01$ with external involvement during stage 3. Effort dedicated to prototyping during stage 3 is positively

CHAPTER 4 • RESULTS

associated at $p < .01$ with the number of engineers and computer scientists involved in both software and prototype development. Further, the extent to which the system is embodied by the prototype at stage 4 is positively associated at $p < .05$ with external involvement across all stages and stage 1.

Forward selection regression was calculated by regressing the independent variables against project duration and schedule overrun using all three project size control variables. The results are shown in Table 8.

Table 8
Results of Forward Selection Regression Analysis

<u>Project Outcome</u>	
<u>Project Duration</u>	
Adjusted R-square	.57
F (significance)	3.19 (.01**)
tb ₁ (significance)	2.75 (.01**)
Functional Involvement Across All Stages	
<u>Schedule Overrun</u>	
Adjusted R-square	.31
F (significance)	1.77 (.14)

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

The results from table 8 indicate that the project duration predictor model accounts for 57% of the variance. Further, a positive association at $p < .05$ was found between project duration and functional involvement across all stages with no project size variables being significant. The schedule overrun predictor model accounts for 31% of the variance.

CHAPTER 4 • RESULTS

No significant associations were found between schedule overrun, the independent variables or the project size variables.

The results of the exploratory analysis indicated that there exists high correlations among the independent variables and linear dependency.

4.7 Results of Regression Analyses for Project Duration and Schedule Overrun

This section present the results of the ordinary least squares regression analyses for those instances where project duration or schedule overrun was used to measure software project outcome. The general form of the regression equations examined was:

$$\text{project outcome} = b_0 + b_1 * \text{Independent variable one} + b_2 * \text{Project size} \quad (\text{Eq. 1})$$

Project outcome was project duration or schedule overrun.

Each of the seven hypotheses had a different independent variable one. The second independent variable was project size. Project size was the control variable used in the regressions. Only one project size variable was used at a time in the model. In addition the model did not use any other explanatory variable other the one identified in the hypothesis. Total code, total cost and total person-months were used to measure project size. However, total cost was excluded since it was strongly associated with total person-months. Calculations were performed with 40 observations for total person-months. Calculations for total code shown in Tables 23, 24, 25 and 26 were performed using 36 observations. All other calculations for total code used 37 observations. One tailed t tests were used since the hypotheses are examining directional associations.

Hypothesis 1

Table 9 shows the regression results relevant to Hypothesis 1: The number of prototypes used throughout the development process is positively associated with software project success.

Table 9
Results of Regression Analysis for Hypothesis 1
(project outcome = $b_0 + b_1 * \text{Number of prototypes} + b_2 * \text{Project size}$)

Project Outcome	Total code	Total person-months
Project Duration		
Adjusted R-square	.05	.33
F (significance)	2.02 (.15)	10.75 (.00***)
tb1 (significance) Number of Prototypes	1.56 (.07)	1.62 (.06)
tb2 (significance) Project Size	.88 (.20)	.53 (.00***)
Schedule Overrun		
Adjusted R-square	.00	.02
F (significance)	.34 (.71)	1.37 (.27)
tb1 (significance) Number of Prototypes	.42 (.34)	.14 (.45)
tb2 (significance) Project Size	.60 (.28)	1.61 (.06)

p<.05; *p<.01

As expected larger projects take longer to complete, schedule overruns are not related to project size. Table 9 shows that no significant association was found between project duration or schedule overrun and the number of prototypes built during software development. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been

CHAPTER 4 • RESULTS

omitted.

The researcher further explored Hypothesis 1 by splitting the data sample depending on whether or not more than one prototype had been used. Regression analysis was performed on the sub-sample of 20 projects for which more than one prototype had been used. The results are shown in Table 10.

Table 10
Results of Regression Analysis for Throwaway Prototypes
(project outcome = $b_0 + b_1 * \text{number of prototypes} + b_2 * \text{Project size}$)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.18	.31
F (significance)	2.87 (.09)	5.25 (.02**)
tb1 (significance) Number of Prototypes	2.39 (.03**)	1.35 (.20)
tb2 (significance) Project Size	-.73 (.48)	1.86 (.08)
<u>Schedule Overrun</u>		
Adjusted R-square	-.10	.00
F (significance)	.24 (.79)	.55 (.59)
tb1 (significance) Number of Prototypes	.68 (.51)	.04 (.97)
tb2 (significance) Project Size	-.09 (.93)	.88 (.39)

p<.05; *p<.01

The results in table 10 show that no significant association at $p<.01$ between project duration and number of prototypes was found.

Hypothesis 2

Table 11 provides the results relevant to Hypothesis 2: The extent to which engineers are involved in both software development and prototype development is positively associated with software project success.

Table 11
Results of Regression Analysis for Hypothesis 2
(project outcome = $b_0 + b_1$ * Number of engineers and computer scientists involved in
both software development and prototype development + b_2 *
Project size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.04	.30
F (significance)	1.75 (.19)	9.33 (.00***)
tb ₁ (significance)	1.39 (.09)	.84 (.21)
Number of Engineers and Computer Scientists Involved in Both Software Development and Prototype Development		
tb ₂ (significance) Project Size	.95 (.09)	3.84 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.02
F (significance)	.25 (.78)	1.41 (.26)
tb ₁ (significance) Number of Engineers	-.01 (.50)	-.31 (.38)
tb ₂ (significance) Project Size	.70 (.25)	1.67 (.05)

p<.05; *p<.01

Table 11 shows that no significant association was found between project duration or schedule overrun and the number of engineers involved in both software development and prototype development. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Hypothesis 3

Tables 12 through 16 provide the results relevant to Hypothesis 3: External involvement in prototype development is positively associated with software project success. Table 12 includes the regression results for when external involvement is measured across all four stages of development. Tables 13, 14, 15 and 16 provide the regression results for external involvement during stages 1, 2, 3 and 4 respectively.

Table 12
Results of Regression Analysis for Hypothesis 3 - Across All Stages
(project outcome = $b_0 + b_1 * \text{sum of involvement of customer employees, suppliers employees and senior executives in prototype development across all stages} + b_2 * \text{Project size}$)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.02	.36
F (significance)	1.41 (.26)	12.15 (.00***)
tb1 (significance) External Involvement Across All Stages	1.12 (.14)	2.13 (.02**)
tb2 (significance) Project Size	1.33 (.10)	4.75 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.09
F (significance)	1.16 (.33)	2.93 (.06)
tb1 (significance) External Involvement Across all stages	1.33 (.19)	1.71 (.09)
tb2 (significance) Project Size	.83 (.41)	1.99 (.05)

p<.05; *p<.01

Table 12 shows that an association significant at $p<.05$ was found between project duration external involvement across all stages when total person-months was used as the control variable. No significant association was found between schedule overrun and the extent of external involvement in prototype development across all stages of software development.

Table 13
 Results of Regression Analysis for Hypothesis 3 - Stage 1
 (project outcome = $b_0 + b_1 * \text{sum of involvement of customer employees, suppliers employees and senior executives in prototype development during stage 1} + b_2 * \text{Project size}$)

<u>Project Outcome</u>	<u>Total code</u>	<u>Total person-months</u>
<u>Project Duration</u>		
Adjusted R-square	.00	.74
F (significance)	.93 (.41)	11.10 (.00***)
tb1 (significance) External Involvement At Stage 1	1.23 (.12)	1.76 (.05)
tb2 (significance) Project Size	.58 (.28)	4.66 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.09
F (significance)	.94 (.40)	3.00 (.06)
tb1 (significance) External Involvement At Stage 1	1.16 (.13)	1.75 (.05)
tb2 (significance) Project Size	.73 (.24)	2.07 (.03**)

p<.05; *p<.01

Table 14
Results of Regression Analysis for Hypothesis 3 - Stage 2
 (project outcome = $b_0 + b_1$ * sum of involvement of customer employees, suppliers
 employees and senior executives in prototype development during stage 2 + b_2 * Project
 size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.00	.30
F (significance)	.75 (.48)	9.22 (.00***)
tb1 (significance) External Involvement At Stage 2	.05 (.49)	.75 (.23)
tb2 (significance) Project Size	1.22 (.12)	4.29 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.03
F (significance)	.29 (.75)	1.58 (.22)
tb1 (significance) External Involvement At Stage 2	.26 (.40)	.63 (.27)
tb2 (significance) Project Size	.68 (.25)	1.73 (.05)
p<.05; *p<.01		

Tables 13 and 14 show that no association significant at $p<.05$ was found between project duration or schedule overrun and external involvement in prototype development during stages 1 and 2 of software development. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Table 15
Results of Regression Analysis for Hypothesis 3 - Stage 3
 (project outcome = $b_0 + b_1$ * sum of involvement of customer employees, suppliers
 employees and senior executives in prototype development during stage 3 + b_2 * Project
 size)

<u>Project Outcome</u>	<u>Total code</u>	<u>Total person-months</u>
<u>Project Duration</u>		
Adjusted R-square	.02	.35
F (significance)	1.38 (.27)	11.30 (.00***)
tb ₁ (significance) External Involvement At Stage 3	1.10 (.14)	1.84 (.04**)
tb ₂ (significance) Project Size	1.17 (.13)	4.51 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.07	.13
F (significance)	2.08 (.14)	3.86 (.03**)
tb ₁ (significance) External Involvement At Stage 3	1.90 (.04)	2.16 (.02***)
tb ₂ (significance) Project Size	.62 (.27)	1.9 (.03**)
p<.05; *p<.01		

Table 15 shows that a positive association significant at $p<.05$ was found between project duration and external involvement when total person-months was used as the control variable. Further a positive association significant at $p<.05$ was found between schedule overrun and external involvement when total person-months was used as the control variable. However, the adjusted R-square statistic for the regression was only .13. The plot of the dependent variable against the residuals in the regression indicated that a

pattern existed. This suggests that important explanatory variables had been omitted.

Table 16
Results of Regression Analysis for Hypothesis 3 - Stage 4
 (project outcome = $b_0 + b_1$ * sum of involvement of customer employees, suppliers
 employees and senior executives in prototype development during stage 4 + b_2 * Project
 size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.11	.39
F (significance)	3.16 (.06)	13.55 (.00***)
tb ₁ (significance) External Involvement At Stage 4	2.15 (.02**)	2.53 (.01**)
tb ₂ (significance) Project Size	2.03 (.03**)	4.86 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.05
F (significance)	1.02 (.37)	2.03 (.15)
tb ₁ (significance) External Involvement At Stage 4	1.23 (.12)	1.11 (.14)
tb ₂ (significance) Project Size	1.15 (.13)	1.82 (.04**)
p<.05; *p<.01		

Table 16 shows that a positive association significant at $p<.05$ was found between project duration and external involvement in prototype development during stage 4 of software development when total person-months was used to measure project size. When total code and total person-months were used as the control variable, the greater the external involvement in prototype development the longer project duration. No association was

CHAPTER 4 • RESULTS

found between schedule overrun and external involvement. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Hypothesis 4

The results relevant to Hypothesis 4 are shown in Tables 17 through 21. Hypothesis 4 states: Functional involvement in prototype development is positively associated with software project success. Table 17 includes the regression results for when functional involvement is measured across all four stages of development.

Table 17
Results of Regression Analysis for Hypothesis 4 - Across All Stages
(project outcome = $b_0 + b_1 * \text{sum of involvement of marketing/sales, product management, software design, customer services, quality and software maintenance in prototype development across all stages} + b_2 * \text{Project size}$)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.00	.35
F (significance)	1.16 (.33)	11.28 (.00***)
tb ₁ (significance) Functional Involvement Across All Stages	.88 (.19)	1.83 (.04**)
tb ₂ (significance) Project Size	1.01 (.16)	4.47 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.09	.17
F (significance)	2.89 (.07)	5.13 (.01**)
tb ₁ (significance) Functional Involvement Across All Stages	2.28 (.02**)	2.65 (.00***)
tb ₂ (significance) Project Size	.23 (.41)	1.94 (.03**)
p<.05; *p<.01		

Table 17 shows that a positive association significant at $p<.05$ was found between project duration and functional involvement in prototype development across all four stages of the project when total person-months was used as the control variable. Further a positive association significant at $p<.05$ was found between schedule overrun and functional involvement in prototype development across all four stages of the project when total person-months was used as the control variable. The plot of the dependent variable

CHAPTER 4 • RESULTS

against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Table 18, 19, 20 and 21 provide the regression results for when functional involvement in prototype development is measured during stage 1, 2, 3 and 4 respectively.

Table 18
Results of Regression Analysis for Hypothesis 4 - Stage 1
 (project outcome = $b_0 + b_1 \cdot \text{sum of involvement of marketing/sales, product management, software design, customer services, quality and software maintenance in prototype development during stage 1}$
 $+ b_2 \cdot \text{Project size}$)

<u>Project Outcome</u>	<u>Total code</u>	<u>Total person-months</u>
<u>Project Duration</u>		
Adjusted R-square	.00	.30
F (significance)	.80 (.46)	9.24 (.00***)
tb1 (significance) Functional Involvement At Stage 1	.32 (.38)	.76 (.23)
tb2 (significance) Project Size	1.01 (.15)	4.22 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.05
F (significance)	.71 (.50)	2.07 (.14)
tb1 (significance) Functional Involvement At Stage 1	.95 (.18)	1.15 (.13)
tb2 (significance) Project Size	.42 (.34)	1.66 (.05)

p<.05; *p<.01

Table 19
Results of Regression Analysis for Hypothesis 4 - Stage 2
 (project outcome = $b_0 + b_1 * \text{sum of involvement of marketing/sales, product management, software design, customer services, quality and software maintenance in prototype development during stage 2} + b_2 * \text{Project size}$)

<u>Project Outcome</u>	<u>Total code</u>	<u>Total person-months</u>
<u>Project Duration</u>		
Adjusted R-square	.00	.30
F (significance)	.76 (.47)	9.20 (.00***)
tb1 (significance) Functional Involvement At Stage 2	.15 (.44)	.73 (.24)
tb2 (significance) Project Size	1.14 (.13)	4.24 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.01	.09
F (significance)	1.20 (.31)	2.88 (.07)
tb1 (significance) Functional Involvement At Stage 2	1.37 (.09)	1.68 (.05)
tb2 (significance) Project Size	.34 (.37)	1.75 (.05)

p<.05; *p<.01

Table 18 and 19 show that no significant association at $p<.05$ was found between project duration or schedule overrun and functional involvement during stages 1 and 2. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Table 20
Results of Regression Analysis for Hypothesis 4 - Stage 3
 (project outcome = $b_0 + b_1 \cdot \text{sum of involvement of marketing/sales, product management, software design, customer services, quality and software maintenance in prototype development during stage 3} + b_2 \cdot \text{Project size}$)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.00	.33
F (significance)	.87 (.43)	10.75 (.00***)
tb ₁ (significance) Functional Involvement At Stage 3	.48 (.32)	1.62 (.06)
tb ₂ (significance) Project Size	1.03 (.16)	4.45 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.12	.20
F (significance)	3.56 (.03**)	5.96 (.01**)
tb ₁ (significance) Functional Involvement At Stage 3	.255 (.00***)	2.93 (.00***)
tb ₂ (significance) Project Size	.00 (.50)	2.05 (.03**)

p<.05; *p<.01

Table 20 shows that a positive association significant at $p<.05$ was found between schedule overrun and functional involvement during stage 3. This result was consistent across both control variables. A similar association between project duration and functional involvement was not found. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Table 21
Results of Regression Analysis for Hypothesis 4 - Stage 4
(project outcome = $b_0 + b_1$ * sum of involvement of marketing/sales, product management, software design, customer services, quality and software maintenance in prototype development during stage 4 + b_2 * Project size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.10	.45
F (significance)	3.08 (.06)	16.97 (.00***)
tb ₁ (significance) Functional Involvement At Stage 4	2.11 (.02**)	3.32 (.00***)
tb ₂ (significance) Project Size	1.39 (.09)	5.07 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.17	.24
F (significance)	4.75 (.01**)	7.25 (.00***)
tb ₁ (significance) Functional Involvement At Stage 4	2.98 (.00***)	3.10 (.00***)
tb ₂ (significance) Project Size	.93 (.18)	2.17 (.02**)

p<.05; *p<.01

Table 21 shows that a positive association significant at $p<.01$ was found between project outcomes and functional involvement during stage 4 of software development. The greater the extent of functional involvement during stage 4, the longer project duration. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Hypothesis 5

Table 22 shows the results relevant to Hypothesis 5: The number of customer sites involved in the assessment of prototypes is positively associated with software project success.

Table 22
Results of Regression Analysis for Hypothesis 5
 (project outcome = $b_0 + b_1 * \text{number of customer sites involved in the assessment of prototypes} + b_2 * \text{Project size}$)

<u>Project Outcome</u>	<u>Total code</u>	<u>Total person-months</u>
<u>Project Duration</u>		
Adjusted R-square	.00	.30
F (significance)	.78 (.46)	9.40 (.00***)
tb1 (significance) Customer Sites	-.26 (.40)	.90 (.19)
tb2 (significance) Project Size	1.21 (.11)	4.34 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	0.00	.02
F (significance)	.28 (.76)	1.47 (.24)
tb1 (significance) Customer Sites	.21 (.42)	.44 (.33)
tb2 (significance) Project Size	.70 (.25)	1.71 (.05)

p<.05; *p<.01

Table 22 shows that no significant association was found between project duration or schedule overrun and the number of customer sites used to assess prototypes. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Hypothesis 6

The results relevant to Hypothesis 6 are shown in Table 23 through 27. Hypothesis 6 states: The amount of effort dedicated to prototyping is positively associated with software project success.

Table 23 includes the regression results for when the amount of effort dedicated to prototyping is measured across all four stages of development.

Table 23
Results of Regression Analysis for Hypothesis 6 - Across All Stages
 (project outcome = $b_0 + b_1$ * the amount of effort dedicated to prototyping across all stages + b_2 * Project size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.13	.31
F (significance)	3.84 (.03**)	9.96 (.00***)
tb1 (significance) Effort Dedicated to Prototyping Across All Stages	2.43 (.01**)	-1.25 (.11)
tb2 (significance) Project Size	1.01 (.14)	3.33 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.03
F (significance)	.77 (.47)	1.50 (.24)
tb1 (significance) Effort Dedicated to Prototyping Across All Stages	1.00 (.16)	-.51 (.31)
tb2 (significance) Project Size	.63 (.27)	1.31 (.10)

p<.05; *p<.01

CHAPTER 4 • RESULTS

Table 23 shows that no significant association was found between project duration or schedule overrun and the amount of effort dedicated to prototyping across all four stages of the project when total person-months was used as the control variable. However, project duration and the amount of effort dedicated to prototyping were found to have an association that was significant at $p < .05$ when total code was used as the control variable. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Tables 24 and 25 provide the regression results for when the amount of effort dedicated to prototyping is measured during stages 1 and 2.

Table 24
Results of Regression Analysis for Hypothesis 6 - Stage 1
(project outcome = $b_0 + b_1 * \text{amount of effort dedicated to prototyping during stage 1}$
+ $b_2 * \text{Project size}$)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.12	.31
F (significance)	3.47 (.04**)	9.12 (.00***)
tb1 (significance) Effort Dedicated to Prototyping At Stage 1	2.12 (.02**)	-.66 (.26)
tb2 (significance) Project Size	1.25 (.11)	3.45 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.02
F (significance)	.55 (.58)	1.35 (.27)
tb1 (significance) Effort Dedicated to Prototyping At Stage 1	.80 (.22)	-.33 (.37)
tb2 (significance) Project Size	.56 (.29)	1.34 (.10)

p<.05; *p<.01

Table 25
Results of Regression Analysis for Hypothesis 6 - Stage 2
 (project outcome = $b_0 + b_1 * \text{amount of effort dedicated to prototyping during stage 2}$
 $+ b_2 * \text{Project size}$)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.02	.43
F (significance)	1.43 (.25)	15.52 (.00***)
tb1 (significance) Effort Dedicated to Prototyping At Stage 2	.83 (.21)	-2.87 (.00***)
tb2 (significance) Project Size	1.30 (.10)	5.35 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.06
F (significance)	.24 (.79)	2.19 (.13)
tb1 (significance) Effort Dedicated to Prototyping At Stage 2	.16 (.44)	-1.30 (.10)
tb2 (significance) Project Size	.62 (.27)	2.07 (.03)
p<.05; *p<.01		

Table 24 shows a significant association at $p<.05$ between project duration and effort dedicated to prototyping during stage 1 when total code is used to measure project size. Table 25 shows a significant association at $p<.01$ between project duration and effort dedicated to prototyping during stage 2 when total person-months is used to measure project size. Tables 24 and 25 did not show a significant association between schedule overrun and the amount of effort dedicated to prototyping during stages 1 and 2. The plot of the dependent variable against the residuals in the regression indicated that a pattern

CHAPTER 4 • RESULTS

existed. This suggests that important explanatory variables had been omitted.

Table 26 shows the regression results from when the amount of effort dedicated to prototyping is measured during stage 3.

Table 26
Results of Regression Analysis for Hypothesis 6 - Stage 3
(project outcome = $b_0 + b_1$ * amount of effort dedicated to prototyping during stage 3
+ b_2 * Project size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.21	.30
F (significance)	5.76 (.00***)	9.29(.00***)
tb ₁ (significance) Effort Dedicated to Prototyping At Stage 3	2.97 (.00***)	-.05 (.48)
tb ₂ (significance) Project Size	1.42 (.08)	2.67 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.02
F (significance)	.79 (.46)	1.36 (.29)
tb ₁ (significance) Effort Dedicated to Prototyping At Stage 3	1.06 (.15)	-.16 (.44)
tb ₂ (significance) Project Size	.60 (.28)	1.11 (.14)

p<.05; *p<.01

Table 26 shows that a positive association significant at $p<.05$ was found between project duration and effort dedicated to prototyping in stage 3 when total code was used to measure project size. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory

CHAPTER 4 • RESULTS

variables had been omitted. Table 27 shows the regression results for when the amount of effort dedicated to prototyping is measured during stage 4.

Table 27
Results of Regression Analysis for Hypothesis 6 - Stage 4
(project outcome = $b_0 + b_1$ * amount of effort dedicated to prototyping during stage 4
+ b_2 * Project size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.07	.34
F (significance)	2.39 (.10)	10.60 (.00***)
t_{b1} (significance) Effort Dedicated to Prototyping At Stage 4	1.57 (.07)	-1.32 (.10)
t_{b2} (significance) Project Size	1.41 (.09)	4.12 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.02
F (significance)	.94 (.40)	1.33 (.28)
t_{b1} (significance) Effort Dedicated to Prototyping At Stage 4	1.19 (.12)	.27 (.40)
t_{b2} (significance) Project Size	.60 (.28)	.98 (.17)

p<.05; *p<.01

Table 27 shows that no significant association was found between project duration or schedule overrun and the amount of effort dedicated to prototyping during stage 4. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted.

Hypothesis 7

Tables 28 through 30 provide the results relevant to Hypothesis 7a, 7b and 7c respectively. Hypothesis 7a states: The extent of the system embodied in the prototypes at the completion of basic software requirements is positively associated with software project success. The results of the regression are shown in Table 28.

Table 28
Results of Regression Analysis for Hypothesis 7a - Stage 1
(project outcome = $b_0 + b_1$ * extent to which the prototypes built at the completion of stage 1 embodied the functionality, real time performance, ease of use and quality of the end-product + b_2 * Project size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.00	.29
F (significance)	.93 (.41)	9.04 (.00***)
tb1 (significance) Extent to Which the Prototypes Embodied the System At the End of Stage 1	-.58 (.28)	.55 (.30)
tb2 (significance) Project Size	1.13 (.14)	4.17 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.05
F (significance)	.544 (.59)	2.09 (.14)
tb1 (significance) Extent to Which the Prototypes Embodied the System At the End of Stage 1	.76 (.23)	1.17 (.13)
tb2 (significance) Project Size	.83 (.21)	1.97 (.03**)

p<.05; *p<.01

CHAPTER 4 • RESULTS

Hypothesis 7b was: The extent of the system embodied in the prototypes at the completion of high level design is positively associated with software project success. The results of the regressions are shown in Table 29.

Table 29
Results of Regression Analysis for Hypothesis 7b - Stage 2
(project outcome = $b_0 + b_1$ * extent to which the prototypes built at the completion of stage 2 embodied the functionality, real time performance, ease of use and quality of the end-product + b_2 * Project size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.00	.29
F (significance)	.91 (.41)	8.94 (.00***)
b_1 (significance) Extent to Which the Prototypes Embodied the System At the End of Stage 2	-.56 (.29)	.41 (.30)
b_2 (significance) Project Size	1.22 (.12)	4.15 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.06
F (significance)	.53 (.59)	2.21 (.12)
b_1 (significance) Extent to Which the Prototypes Embodied the System At the End of Stage 2	.74 (.24)	1.26 (.11)
b_2 (significance) Project Size	.73 (.24)	1.97 (.03**)

p<.05; *p<.01

CHAPTER 4 • RESULTS

Hypothesis 7c was: The extent of the system embodied in the prototypes at the start of field testing of the software is positively associated with software project success. The results of the regressions are shown in Table 30.

Table 30
Results of Regression Analysis for Hypothesis 7c - Stage 3
(project outcome = $b_0 + b_1$ * extent to which the prototypes built at the start of stage 3 embodied the functionality, real time performance, ease of use and quality of the end-product + b_2 * Project size)

Project Outcome	Total code	Total person-months
<u>Project Duration</u>		
Adjusted R-square	.01	.31
F (significance)	.80 (.46)	9.61 (.00***)
tb1 (significance) Extent to Which the Prototypes Embodied the System At the Start of Stage 3	-.30 (.39)	1.04 (.15)
tb2 (significance) Project Size	1.26 (.11)	4.38 (.00***)
<u>Schedule Overrun</u>		
Adjusted R-square	.00	.03
F (significance)	.26 (.77)	1.60 (.21)
tb1 (significance) Extent to Which the Prototypes Embodied the System At the Start of Stage 3	.08 (.47)	.68 (.25)
tb2 (significance) Project Size	.68 (.25)	1.78 (.04**)
p<.05; *p<.01		

The results in Tables 28, 29 and 30 show that no relationship significant at $p<.05$ between project duration or schedule overrun and the extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the software

CHAPTER 4 • RESULTS

was found. The plot of the dependent variable against the residuals in the regression indicated that a pattern existed. This suggests that important explanatory variables had been omitted. Further, the relationships found were weak and contrary to the hypothesized direction.

4.8 Results of Correlation Analyses for Project Manager's Satisfaction that the Software Met Customer Requirements

This section presents the results of the zero-order correlation analyses for those instances where project manager's satisfaction that the software met customer requirements was used as measures of project outcome. Table 31 shows the correlations between the independent variables identified in section 4.2 and the one measure of project outcome.

Table 31
Correlations for Project Manager's Satisfaction that the Software Met Customer Requirements

Independent Variable	Project manager's satisfaction
Number of prototypes	.00
Number of engineers involved in both software development and prototype development	-.04
Sum of involvement of customers' employees, suppliers' employees and senior executives in prototype development	
• across all stages	.07
• during stage 1 (between key points A and B)	.10
• during stage 2 (between key points B and C)	.08
• during stage 3 (between key points C and D)	-.02
• during stage 4 (between key points D and E)	.07
Sum of involvement of marketing/sales, product management, software design, customer services, quality and software maintenance in prototype development	
• across all stages	.07
• during stage 1 (between key points A and B)	.15
• during stage 2 (between key points B and C)	.11
• during stage 3 (between key points C and D)	.01
• during stage 4 (between key points D and E)	.01
Number of customer sites involved in the assessment of prototypes	.14
Effort dedicated to prototyping:	
• across all stages (person-months)	.00
• during stage 1 (between key points A and B) (person-months)	.00
• during stage 2 (between key points B and C) (person-months)	.06
• during stage 3 (between key points C and D) (person-months)	.09
• during stage 4 (between key points D and E) (person-months)	-.05
Extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the product	
• at the end of stage 1	.26
• at the end of stage 2	.39***
• at the start of stage 3	.50***

p<.05; *p<.01

The results in Table 31 show a strong relationship between the project manager's satisfaction that the software met customer requirements and the extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the

product at the end of stage 2 and the start of stage 3.

The results in table 31 were calculated using 39 observations for the amount of effort dedicated to prototyping during stages 1, 2, 3 and 4. All other calculations were performed with 40 observations.

4.9 Summary of Significant Results from Ordinary Least Squares Regression and Zero-Order Correlations

Table 32 highlights the associations between project outcomes and independent variables that were found to be significant at $p < .05$ and $p < .01$.

The results of the ordinary least squares controlled for only one project size variable and as such each model was a weak predictor model. Further each model was unable to control the influence from other variables since some of the variables are missing in this study. This might explain the unexpected results generated in section 4.6. In addition, residual analysis indicated the presence of patterns in the data. As a result the researcher was led to look at more complex models.

Table 32
Summary of Significant Results From Ordinary Least Squares Regression and Zero-Order Correlations

Hypothesis	Project duration	Schedule overrun	Project manager's satisfaction
1. Number of prototypes			
2. Number of engineers involved in both software and prototype development			
3. External involvement	across all stages sig. at .05 using total person-months	for stage 3 sig. at .05 using total person-months	
	for stages 3 and 4 sig. at .05 using total person-months		
4. Functional Involvement	across all stages sig. at .05 using total person-months	across all stages sig. at .01 using total person-months	
	for stage 4 sig. at .01 using total person-months & sig. at .05 using total code	for stage 3 sig. at .05 for both size variables	
		for stage 4 sig. at .01 using total person-months & sig. at .05 using total code	
5. Number of customer sites involved in the assessment of prototypes			
6. Effort dedicated to prototyping (person-months)	across all stages sig. at .05 using total total code		
	for stage 1 sig. at .05 using total code		
	for stages 2 and 3 sig. at .01 using total code		
7. Extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the product			for stages 2 and 3 sig. at .01

4.10 Results of Stepwise Regression Analyses for Project Duration and Schedule Overrun

To address the fact that the relationships found in section 4.6 are opposite to those hypothesized and to address the fact that the models used in section 4.6 were based only on a subset of variables and showed poor fits in general, it was decided to undertake a stepwise regression analysis of the data.

This section presents the results of this stepwise regression analyses for those instances where project duration and schedule overrun were used to measure software project outcome.

Stepwise regression was chosen to develop a model that would minimize the influence of missing variables and maximize the predictiveness of the variables used in this study. Two stepwise regressions were run for each of the dependent variables, project duration and schedule overrun. All the independent and control variables were entered into the stepwise regression. One-tailed t tests were used since the hypotheses are examining directional associations.

Tables 33 and 34 show the results of the stepwise regression analysis used to develop the optimum predictor models for both project duration and schedule overrun.

The final project duration predictor model was the following:

project duration = $b_0 - b_1 * \text{number of engineers and computer scientists involved in both software development and prototype development} - b_2 * \text{functional involvement across all stages} + b_3 * \text{functional involvement at stage 4} + b_4 * \text{extent to which the prototypes embodied the system at the start of stage 3} + b_5 * \text{total cost}$ (Eq. 2)

The final schedule overrun predictor model was the following:

$$\text{schedule overrun} = b_0 + b_1 * \text{functional involvement at stage 4} \quad (\text{Eq. 3})$$

Table 33
Results of Stepwise Regression Analysis for Project Duration
(project duration = $b_0 + b_1$ * number of engineers and computer scientists involved in both software development and prototype development + b_2 * functional involvement across all stages + b_3 * functional involvement at stage 4 + b_4 * extent to which the prototypes embodied the system at the start of stage 3 + b_5 * total cost)
(number of cases = 40)

Project Outcome	Total cost
<u>Project Duration</u>	
Adjusted R-square	.67
F (significance)	15.44 (.00***)
b_1 (se b_1)	1.34 (1.50)
tb_1 (significance)	2.89 (.00***)
Number of Engineers and Computer Scientists Involved in Both Software and Prototype Development	
b_2 (se b_2)	4.61 (1.53)
tb_2 (significance)	3.01 (.00***)
Functional Involvement Across All Stages	
b_3 (se b_3)	27.36 (5.54)
tb_3 (significance)	4.94 (.00***)
Functional Involvement At Stage 4	
b_4 (se b_4)	.37 (.14)
tb_4 (significance)	2.66 (.01**)
Extent to Which the Prototypes Embodied the System at the Start of Stage 3	
b_5 (se b_5)	4.67E-05 (6.19E-06)
tb_5 (significance) Total Cost	7.60 (.00***)

p<.05; *p<.01

Table 34
Results of Stepwise Regression Analysis for Schedule Overrun
(schedule overrun = $b_0 + b_1 \cdot \text{functional involvement at stage 4}$)
(number of cases = 40)

<u>Project Outcome</u>	
<u>Schedule Overrun</u>	
Adjusted R-square	.19
F (significance)	9.46 (.00)***
b_1 (se b_1)	8.38 (2.73)
tb_1 (significance)	3.08 (.00)***
<u>Functional Involvement At Stage 4</u>	
** $p < .05$; *** $p < .01$	

Tables 33 and 34 show the optimum predictor models developed for both project duration and schedule overrun. The project duration predictor model can account for 67% of the variance in project duration using four independent variables controlling for total cost. There is a positive association at $p < .01$ between project duration and the number of engineers and computer scientists involved in both software and prototype development. There is a positive association at $p < .01$ between functional involvement across all stages of software development and functional involvement during stage 4. However, it should be noted that functional involvement across all stages of software development and functional involvement during stage 4 are correlated (See Appendix D). Excluding functional involvement across all stages from the stepwise regression analysis did not improve the results shown in Table 33.

There is a positive association at $p < .01$ between the extent to which the prototypes embodied the system after the high level design of the software is completed and project

duration.

Standardized residual analysis does indicate that there is a linear association between project duration and the standardized residual. This is due to a missing explanatory variable that has a linear association with project duration. More research needs to be conducted to identify those variables that may account for the remaining 23% of the variance.

The optimum schedule overrun predictor model developed from the stepwise regression can only account for 19% of the variance in schedule overrun. There is a positive correlation at $p < .01$ between functional involvement during stage 4 and schedule overrun. Standardized residual analysis does indicate that there is a linear association between schedule overrun and the standardized residual. This is due to a missing explanatory variable that has a linear association with schedule overrun. More research needs to be conducted to identify those variables that may account for the remaining 81% of the variance. Standardized residuals and casewise plots were used to identify outliers that may exist in both predictor models. Both standardized residuals and the casewise plot identified case 7 as an outlier in the project duration model (re. it was poorly predicted by the stepwise model). Schedule overrun did not have any outliers identified. It should be noted that case 7 was not used in the original stepwise model calculation because it involved missing variable values and hence its removal did not change the model. Therefore, a second stepwise regression was not run. The respondents for case 7 were unable to provide data on the proportion of total person-months dedicated to the development and assessment of prototypes. The respondent considered this information confidential.

2

of/de

2

PM-1 3 1/2" x 4" PHOTOGRAPHIC MICROCOPY TARGET
NBS 1010a ANSI/ISO #2 EQUIVALENT

PRECISIONSM RESOLUTION TARGETS

4.11 List of Important Prototype Management Decisions Made to Shorten Project Duration, Schedule Overrun and Increase Project Manager's Satisfaction that the Software Met Customer Requirements

This section lists the responses to questions 4.1, 4.2 and 4.3 of the questionnaire. The respondents were asked to list the most important prototype management decisions that contributed to a reduction in project duration, schedule overrun and the project manager's satisfaction that the software met customer requirements. The researcher selected five categories to organize the presentation of the responses: timely resource allocation, incorporation of stakeholders concerns into the design, development process characteristics, maintenance of schedule and product vision and experience.

4.11.1 Important Prototype Management Decisions that Contributed to a Reduction in Project Duration

Timely Resource Allocation

1. making available key resources (experienced personnel and money) throughout the software development process
2. having full-time employees allocated to prototype development

Incorporation of Stakeholders Concerns Into the Design

1. having early involvement of the customer and marketing in the software development process
2. using the prototype to create a shared view of the problem domain amongst multi-functional product development team members

CHAPTER 4 • RESULTS

3. using the prototype as a tool to catch and solve difficult and costly design problems, such as the integration of new customer specifications / requirements concerning, ease of use, functionality, quality, real-time performance or the GUI, early before they become costly to change
4. defining customer requirements early
5. develop as many high quality prototypes as soon as possible

Development Process Characteristics

1. using the prototype to show (for clarification purposes) the customer a version of the product based upon their specifications / requirements
2. using the prototype to test new technology or tools
3. not changing the basic application
4. defining the main purpose or function of the prototype in the early stages of the software development process. For example, is the prototype to be used to map out specifications or could it be implemented in the software development process with very little modifications.
5. reusing as much of the prototype (in terms of code or parts) as possible when developing other prototype iterations or the end product
6. concentrating on the core product requirements and leaving the peripheral functionality to the end

Maintenance of Schedule and Product Vision

1. using the prototype as means to track the progress of the product through the software development process

Team Member Experience

1. having managers trained in the prototype process

One respondent indicated that an effective way to reduce project duration was to not prototype at all. According to this respondent prototyping lengthens project duration.

4.11.2 Important Prototype Management Decisions that Contributed to a Reduction in Schedule Overrun

Timely Resource Allocation

1. making available financial resources
2. having full-time employees involved in the prototype's development

Incorporation of Stakeholders Concerns Into the Design

1. involving users early so that they can get "hands-on experience" with the prototype
2. having the engineers work directly with the customer
3. nailing down as much of the user interface requirements as soon as possible
4. having marketing involved early in the software development process

Development Process Characteristics

1. using well-known software development tools and languages
2. developing evolutionary prototypes early in the software development process
3. partitioning tasks into smaller more manageable tasks with frequent development

CHAPTER 4 • RESULTS

intervals

- 4. prototyping the hardware / software interface to reduce the risk of integration problems**
- 5. forcing the customer to adhere to the installation date of the prototype**
- 6. using the prototype as a tool to catch design errors early so that they do not become expensive to change. This allows a reduction in the software development cycle time.**
- 7. collecting client commitment in advance to key components of the infrastructure and scope of the project**
- 8. having the interface rules established early, with all parties in agreement to avoid time and cost changes later in the software development process**
- 9. having progressive "builds" with increased functionality**
- 10. working on the "core" of the system first then adding the "bells and whistles" at the end**
- 11. reusing an existing product platform to develop the prototype**
- 12. using the pressure of developing the product on time**
- 13. not changing the basic application**
- 14. converting the software at the end of the software development process. For example converting Oracle form 3.0 to Oracle form 4.0**
- 15. going directly from the prototype to the end product, since the prototype met most of the product goals**
- 16. ceasing prototyping and starting to perform quality assurance on what you have built**
- 17. reducing the number of features in the product**
- 18. defining product requirements thoroughly**

Maintenance of Schedule and Product Vision

1. documenting and controlling thoroughly the project's progress for any slippage
2. defining the main purpose of the prototype early in the software development process

Team Member Experience

1. having experienced prototype personnel on the product development team

4.11.3 Important Prototype Management Decisions that Contributed to An Increase in the Project Manager's Satisfaction that the Software Met Customer Requirements

Incorporation of Stakeholders Concerns Into the Design

1. involving as many users as possible
2. having direct involvement of the end user in all phases of the software development process (requirements, tool selection, and building of the prototype)
3. asking initial users for marketability input from what was termed "the demo". From their responses the "visual metaphor (look and feel)" necessary to begin full production can be established.
4. getting marketing involved early in the software development process
5. having full-time participation of the user representative
6. involving the customer early in the software development process
7. having a staff member who is an end user and let her use each prototype and implement her feedback
8. having repeated demonstrations to customers and receiving immediate feed-back

CHAPTER 4 • RESULTS

9. involving all customers at user group meetings and getting feedback from customers during a range of stages throughout their purchasing and testing periods of current software
10. letting users see the user-interface mock-up

Development Process Characteristics

1. having a long alpha and beta test drive at various work areas and on different networks
2. maintaining comprehensive documentation of required features for comparison with delivered product
3. using early prototypes of new user interfaces to discard confusing and difficult implementation problems
4. using prototypes to find design problems
5. using prototypes to find real-time constraints, i.e., what is the critical path
6. having defined specifications and functionality very early and quickly in the software development process
7. being able to trace all design and test data to the requirements
8. releasing the product on time and having it meet, preferably exceed, customer expectations
9. making the primary item at review meetings with customers the demonstration of the prototype's functionality
10. using robust testing methods
11. not changing basic application
12. concentrating only on the user interface, i.e., GUI
13. starting field testing (beta release to the customer) of the prototype early in the software

CHAPTER 4. RESULTS

development process

Maintain Schedule and Product Vision

1. having the prototype meet most of the important goals

Team Member Experience

1. having expert subject matter knowledge

CHAPTER 5

DISCUSSION OF RESULTS

This chapter discusses the results of the study. Section 5.1 presents a summary of the results of this study. Sections 5.2 to 5.6 highlight the most interesting results of the study. Section 5.7 discusses the limitations of the research.

5.1 Discussion of Results

This is the first empirical study of the relationship between software project outcomes and prototype management practice. Seven hypotheses were developed from the literature on product development and software management and then tested using a sample of 40 projects drawn from firms developing software in Canada. Several of the results are worth highlighting.

First, the results suggest that, in the software development setting, increasing the number of engineers and computer scientists involved in both software and prototype development reduces project duration.

Second, increased functional involvement across all stages of software development reduces project duration.

Third, the results suggest that, in the software development setting, project duration and schedule overrun are positively associated with the extent of functional involvement in the prototype construction and assessment that occurs during the later stages of software development. Increasing the number of functions involved in prototyping after the high level design of the software has been completed by one will result in an increase in project duration of approximately 27 weeks and an increase in schedule overrun of approximately

8 weeks.

These results are consistent with the literature that argues that internal involvement in prototyping should occur during the early stages of software development and not after the high level design of the software has been completed.

Fourth, the extent to which the system is embodied in the prototype after the high level design of the software has been completed is associated with project duration. This suggests that project duration will increase for projects where prototyping took place during stage 4 and at the completion of stage 3 the prototype represented the entire system.

Fifth, project manager's satisfaction that the software met customer requirements was found to be associated with the extent to which the prototype embodied the characteristics of the end-product at the completion of the software's high level design.

Sixth, the results also show that project duration, schedule overrun and project manager's satisfaction are not linked to the number of prototypes used throughout the development process, external involvement, the number of customer sites used to assess the prototypes and the amount of effort dedicated to prototyping. This finding is consistent with the argument that prototyping does not lead to schedule overruns.

Seventh, decisions that contributed to shorten project duration, reduce schedule overrun and increase project manager's satisfaction with the software can be grouped into: i. timely resource allocation, ii. incorporation of stakeholders' concerns into the design, iii. development process characteristics, iv. maintenance of schedule and product vision and iv. experience of team members.

Eighth, tools and languages are two important components of prototype development environments. Tools used to prototype can be classified by whether or not they implement the object-oriented approach to development and by degree of

specialization. A variety of languages are used to prototype. Most of these languages are programming languages that implement the object-oriented approach to development.

5.2 Integrate Software and Prototype Development Teams

The results of this research emphasize the benefits of having engineers and computer scientists involved in both software and prototype development. The larger the number of engineers and computer scientists who are members of both software and prototype development team(s) the shorter project duration.

Integrating software and prototype development teams will increase problem solving capabilities much in the same way as does the integration between manufacturing and design. Integration leads to the development of a common language and an increase in the level of communication.

5.3 Involve Functions in Prototyping Early

The results of this research indicate that increasing the involvement of one function across all stages of software development will reduce project duration by approximately 4.5 weeks. The finding that functional involvement across all stages of software development will reduce project duration supports arguments made by Barkin and Iansiti (1994); Gupta and Wilemon (1990) and Wheelwright and Clark (1992).

However, an increase of one function in the development and assessment of prototypes after the high level design of the software has been completed will increase project duration by approximately 27 weeks and schedule overrun by approximately 8 weeks. This finding is supportive of Barkin and Iansiti (1994); Gopalan (1995); Gupta and Wilemon (1990) and Wheelwright and Clark (1992) who claimed that late involvement

of functional team members will invariably cause an increase in project duration and schedule overrun.

Project managers need to use prototypes as a tool to gather, assimilate and package all information of the problem domain into a prototype before high level design of the software has been completed. This will allow all function's to focus their specialized knowledge on the problem domain and at the same time use the prototype to translate their knowledge seamlessly to the other team members. Project managers that use prototypes in this manner will reduce the proliferation of prototypes and multiple views developed by each participant before the high level design of the software is completed.

5.4 Extent that the Product is Embodied in the Prototype

The results of this research emphasize that there is a need to define what are the critical system issues that should be embodied by the prototype at each stage of the software development process. The results suggest that project duration will increase if prototyping is continued after the high level design of the software is completed. This finding suggests that manager's must develop many functional prototypes early in the development process up to stage 3 (begins with the completion of the software's high level of the design). If prototypes are needed after stage 3 then the project manager and the development team may not have prototyped correctly ie. customer specifications were not specific enough early in the software development process. This supports the findings by Bischofberger and Pomberger (1992), Barkin and Iansiti (1994), Clark and Fujimoto (1991), Gordon and Bieman (1995), Hardgrave (1995), Leonard-Barton (1991) and Maras *et al.* (1992). These researchers found that the more the prototype embodies the end product early in the development process the sooner robust tests can be performed. This

reduces project duration.

The extent to which the prototype embodied the systems functionality, real-time performance, ease of use and quality at the completion of the software's high level design was significantly associated with the project manager's satisfaction that the software met customer requirements. This supports Bischofberger and Pomberger (1992); Barkin and Iansiti (1994); Clark and Fujimoto (1991); Gordon and Bieman (1995); Hardgrave (1995); Leonard-Barton (1991) and Maras *et al.* (1992) who found that the more the prototype embodies the final product early in the development process the easier it is to detect costly design errors early.

5.5 Use Prototypes to Document and Track the Progress of the Project

The results of this research emphasize the need for project manager's to understand the critical decision making processes that allowed them to shorten project duration, schedule overrun and increase the project manager's satisfaction that the software met customer requirements. Prototypes allow project manager's to closely track the progress of a project for any possible schedule overruns or resource shortages. However, to do this manager's need to become more active in the prototyping process. A better understanding of the prototyping process allows a manager to develop a library of knowledge on the important management decisions that led to a shortening of project duration, a reduction in schedule overrun and an increase in their satisfaction that the software met customer requirements.

The majority of respondents indicated that the most important prototype management decision that contributed to a reduction in project duration was contingent upon the incorporation of stakeholders concerns into the design. "Having early

involvement of customer and marketing personnel" and "defining the customer requirements early" were some of the respondent comments. These findings are consistent with (Barkin & Iansiti, 1994) who argue that these external stakeholders should be involved as early as possible in the prototyping process. The primary objective of external involvement is to detail and validate the product's specifications while the cost of changes in specifications is low (Budde *et al.*, 1992; Lichter *et al.*, 1994; Ulrich & Eppinger, 1995; Vonk, 1990 and Wheelwright & Clark 1992).

The majority of respondents indicated that the most important prototype management decision that contributed to a reduction in schedule overrun was contingent upon the development process characteristics. "Using the prototype as a tool to catch design errors so that they do not become expensive to change" and "having progressive builds with increased functionality" were some of the respondent comments. These findings are consistent with (Barkin & Iansiti, 1994; Clark & Fujimoto, 1991; Wheelwright & Clark, 1992) who argue that early prototypes allow for the faster detection of costly design errors. Further, (Barkin & Iansiti, 1994) found that the best time to rapidly produce high quality prototypes is in the first 15 percent of the development project. This "window of opportunity" spans from product definition; concept selection; process definition to just before the layout stage (Barkin & Iansiti, 1994: 221, Table 7.2).

The majority of respondents indicated that the most important prototype management decision that contributed to an increase in the project manager's satisfaction that the software met customer requirements was contingent upon the development process characteristics. "Start field testing (beta release to the customer) of the prototype early in the software development process" and "using robust testing methods" were some of the respondent comments. These findings are supported by Gordon and Bieman (1995) found

CHAPTER 5 • DISCUSSION OF RESULTS

that user involvement in the requirements definition and testing of in-house prototypes has a positive effect on the product by increasing the likelihood that user needs will be met (Gordon & Bieman, 1995: 92). Vonk (1990) indicates that allowing customers to assess a field prototype strongly influences the success of the prototyping process (Vonk, 1990: 90). Lichter *et al.* (1994) found that the involvement of real-end users is of major importance for the success or failure of the prototyping project (Lichter *et al.*, 1994: 830). Most advocates of prototyping point to user involvement as one of the major benefits provided by prototypes (Clark & Fujimoto, 1991; Lichter *et al.*, 1994 and Wheelwright & Clark, 1992).

5.6 Prototyping Tools and Languages

The results of this research suggest that there are no industry standard tools and languages to develop prototypes. The selection of tools and languages to develop prototypes will be contingent upon a variety of factors such as real-time performance, ease and interoperability with other tools and languages.

Engineers must be able to use prototyping tools and languages to rapidly produce highly representative prototypes that are easy to modify. To Hardgrave (1995), these tools and languages must provide fast response time to user requests and allow the prototypes to evolve into the final product (Hardgrave, 1995: 116).

Prototyping tools and languages must be easy to use. Budde *et al.* (1992); Gordon and Bieman (1995) and Lichter *et al.* (1994) have found that having good tools and languages is only one way of accelerating the prototype development process.

Further, the tools and languages must be fully integrated into the prototyping process, be compatible with other toolsets and languages and be understood by the

developers. Poor integration of the various tools reduces the learning from one prototype iteration to the next (Budde *et al.*, 1992: 164).

5.7 Limitations of Research Approach

There are four limitations to the research approach taken. First, the approach was cross-sectional. A longitudinal research design would have enabled the researcher to examine the other dimensions of prototype management. However, time and cost constraints did not permit the researcher to follow software projects from start to finish.

Second, the methodology used involved one respondent for each project. Many of their responses are subjective and retrospective. 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.

Third, sample size limited the forms of statistical data analysis available. Again, this was an issue of time and cost.

Fourth, the distributions of some of the independent variables (number of prototypes used throughout the development process, number of customer sites involved in the assessment of prototypes, effort dedicated to prototyping and the extent to which the prototypes embodied the system at the end of stage 1) violated the normality assumption required to perform ordinary least squares regression.

REFERENCES

REFERENCES

- Barkin, P., & Iansiti, I. 1993. Concurrent engineering: Research and application. In H.K. Bowen, K.B. Clark, C.A. Holloway, & S.C. Wheelwright (Eds.), The perpetual enterprise machine: Seven keys to corporate renewal through successful product and process development: 202-228. New York: Oxford University Press.
- Bischofberger, W., & Pomberger, G. 1992. Prototyping-Oriented software development: concepts and tools. Berlin, Germany: Springer-Verlag.
- Blum, B.Z. 1986. Application systems prototyping. In M.E. Lipp (Ed.), Prototyping-State of the art report: 3-14. New York: Pergamon Infotech Ltd.
- Boar, B.H. 1984. Application prototyping-A requirements definition strategy of the 80's. New York: John Wiley & Sons.
- Boehm, B.W. 1976. Software engineering economics. IEEE Transactions On Computers, C-25(12): 1226-1241
- Boehm, B.W. 1981. Software engineering economics. New York: Prentice-Hall.
- Boehm, B.W., Gray, T.E., & Seewaldt, T. 1984. Prototyping versus specifying: A multi-project experiment. IEEE Transactions on Software Engineering, 10(3): 290-303.
- Budde, R., Kautz, K., Kuhlenskamp, K., & Züllighoven, H. 1992. Prototyping: An approach to evolutionary system development. Heidelberg, Germany: Springer-Verlag.
- Carey, J.M., & Currey, J.D. 1989. The prototyping conundrum. Datamation, 35(11): 29-33.
- Carey, J.M., & McLeod, R. 1988. Use of system development methodology and tools. Journal of Systems Management, 39(3): 30-35.
- Clark, K.B., & Fujimoto, T. 1991. Product development performance: Strategy, organization and management in the world auto industry. Boston, M.A: Harvard Business School Press.
- Connel, J.L. & Schafer, L.B. 1989. Structured rapid prototyping. Englewood Cliffs, NJ: Yourdon Press
- Cummings, T.G., & Worley, C.G. 1993. Organization development and change. Minneapolis, St. Paul: West Publishing Company.
- Davis, A.M. 1992. Operational prototyping: A new development approach. IEEE Software, 9: 70-78.

REFERENCES

- Doke, E.R. 1990. An industry survey of emerging prototyping methodologies. Information and Management, 18: 169-176.
- Eisenhardt, K.M., & Tabrizi, B.N. 1995. Accelerating adaptive processes: Product innovation in the global computer industry. Administrative Science Quarterly, 40(1): 84-110.
- Gladstein, D. 1984. Groups in context: A model of task group effectiveness. Administrative Science Quarterly, 29(4): 499-517.
- Gopalan, S. 1995. Reasons for schedule overruns in software development projects. Unpublished master's thesis, Carleton University, Canada.
- Gordon, V.S., & Bieman, J.M. 1995. Rapid prototyping: Lessons learned. IEEE Software, 12: 85-95.
- Griffen, A., & Page, A.L. 1993. An interim report on measuring product development success and failure. Journal of Product Innovation Management, 10: 291-306.
- Gupta, A.K., & Wilemon, D. 1988. The credibility-Cooperation connection at the R&D - Marketing interface. Journal of Product Innovation Management, 5(1): 20-31.
- Gupta, A.K., & Wilemon, D.L. 1990. Accelerating the development of technology-based new products. California Management Review, 32(2): 24-44.
- Hardgrave, B.C. 1995. When to prototype: Decision variables used in industry. Information and Software Technology, 37(2): 113-118.
- Hise, R.T., O'Neal, L., Parasuraman, A., & McNeal, J.U. 1990. Marketing / R&D interaction in new product development: Implications for new product success rates. Journal of Product Innovation Management, 7: 142-155.
- Jenkins, A.M., Naumann, J.D., & Wetherbe, J.C. 1984. Empirical investigation of systems development practices and results. Information and Management, 7: 73 - 82.
- Länge, G.B., Leitheiser, R.L., & Naumann, J.D. 1984. A survey of application system prototyping in industry. Information and Management, 7: 273 -284.
- Leonard-Barton, D. 1991. Inanimate integrators: A block of wood speaks. Design Management Journal, 2(3): 61-67.
- Lichter, H., Schneider-Hufschmidt, M., & Züllighoven, H. 1994. Prototyping in industrial software projects-bridging the gap between theory and practice. IEEE Transactions On Software Engineering, 20(11): 825-832.
- Luqi & Royce, W. 1991. Status report: Computer-Aided prototyping. IEEE Software, 5: 77-81.

REFERENCES

- Maras, M.G., Riel, E.J., Nowlan, D.R., Menzel, L.J., & Mahoney, M.J. 1994. A rapid prototyping and integrated design system (RAPIDS) for software development of GN & C systems. Advances in the Astronautical Sciences, 86: 89-102.
- McDonough III, E.F. 1993. Faster new product development: Investigating the effects of technology and characteristics of the project leader and team. Journal of Product Innovation Management, 10: 241-250.
- Nadler, D.A., & Tushman M.L. 1992. Formal organization arrangements: Structure and systems. In Tushman M.L., C. O'Reilly, & D.A. Nadler (Eds.), The management of organizations: 137-153. New York: Harper.
- Necco, C.R., Gordon, C.L., & Tsai, N.W. 1987. Systems analysis and design: Current practices. MIS Quarterly, 11(4): 461-475.
- Paulish, D.J., & Carleton, A.D. 1994. Case studies of software-process-improvement measurement. IEEE Computer, 27(9): 50-57
- Pisano, G.P. 1994. Knowledge, integration and the locus of learning: An empirical analysis of process development. Working Paper No. 95-006. Harvard Business School, Cambridge, MA.
- Souder, W.E. 1988. Managing relations between R&D and marketing in new product development projects. Journal of Product Innovation Management, 5(1): 6-19.
- Ulrich, K.T., & Eppinger, S. D. 1995. Product design and development. New York: McGraw-Hill.
- United States of America Department of Defense. 1988. Military standard: Defense system software development. DOD-STD-2167A. Washington, D.C.
- van Genuchten, M. 1991. Why is software late? An empirical study of reasons for delay in software development. IEEE Transactions on Software Engineering, 17(6): 582-590.
- Vonk, R. 1990. Prototyping: The effective use of case technology. New York: Prentice Hall.
- Wall, M.B., Ulrich, K.T., & Flowers, W.C. 1991. Evaluating prototyping technologies for product design. Working Paper No. 3334-91-MSA, Sloan School of Management, MIT, Cambridge, MA.
- Wheelwright, S.C., & Clark, K.B. 1992. Revolutionizing product development performance: Quantum leaps in speed, efficiency and quality. New York: The Free Press.

APPENDIX A

**Appendix A
Cover Letter Sent to Presidents or Vice Presidents of Selected Companies**

September 20, 1996

NAME AND ADDRESS OF PRESIDENT OR VICE PRESIDENT SELECTED

Dear _____

I am seeking your cooperation in the collection of data for my master's thesis entitled "An Investigation of the Dimensions of Prototype Management that Affect the Outcomes of Software Development Projects". This thesis is being supervised by Professor A.J. Bailetti of the School of Business and the Department of Systems and Computer Engineering.

I need two things from your organization. First, I need you to identify up to three software development projects completed in your organization since October, 1992. Each of the projects you identify must have used one or more prototypes to develop either a custom system, a semi-custom system or a standard software product. I will randomly select one of the projects you identify for inclusion in my sample. Second, a half hour for the project manager responsible for the selected project to complete the questionnaire attached to this letter. All information provided to me will be kept confidential. In my thesis, no link will be made between you, the project manager, your company and the data provided to me.

I would very much appreciate your cooperation. Please provide me with the names of the three software development projects either by mail, fax/voice (788-2660) or e-mail (fsmith@business.carleton.ca) at your earliest convenience. I will contact you shortly afterwards to obtain the name of the project manager to whom the attached letter and questionnaire should be sent.

Should you require any clarification please do not hesitate to contact me.

Sincerely,

Fraser Mac Connell Smith, Graduate Student
Software Technology Management Group

APPENDIX B

**Appendix B
Cover Letter Sent to Manager of Project Selected**

September 20, 1996

NAME AND ADDRESS OF MANAGER SELECTED

Dear _____

I am seeking your cooperation in the collection of data for my master's thesis entitled "An Investigation of the Dimensions of Prototype Management that Affect the Outcomes of Software Development Projects". This thesis is being supervised by Professor A.J. Bailetti of the School of Business and the Department of Systems and Computer Engineering.

I need you to complete the questionnaire that is attached to this letter. This will take less than 30 minutes. All information provided to me will be kept confidential. In my thesis, no link will be made between you, your company and the data provided to me. The information that is sought pertains to xxxxx, a software development project which you managed. The project was included in the sample because prototyping was used to develop the software.

I would very much appreciate your participation in this study. You can return the completed questionnaire to me either by mail or fax (788-2660).

I have tried to make the questionnaire as user-friendly as possible. However, should you require any clarification please do not hesitate to contact me by voice/fax (788-2660) or e-mail (fsmith@business.carleton.ca). I encourage you to consult with others in your organization when providing the information requested in the questionnaire so as to ensure data accuracy.

Sincerely,

**Fraser Mac Connell Smith, Graduate Student
Software Technology Management Group**

APPENDIX C

Appendix C Prototype Management Questionnaire

INSTRUCTIONS

This questionnaire focuses on the use of prototyping in the xxxxx project. By a prototype I mean a working model that embodies some or all the aspects of the end-product. Prototypes can be used to show the feasibility of the future application system, demonstrate the functionality of the user interface, illustrate specific aspects of a problem, develop the core of the application system and so on.

Completing the questionnaire is expected to take less than 30 minutes. Some of the questions require that stages of the development process be identified. There is no widely accepted sequence of stages for software development, thus I have partitioned the development cycle using key points A, B, C, D and E as defined in the following diagram. Please indicate the dates for each key point of the xxxxx project. This will facilitate the completion of the questionnaire.

If you have difficulty answering a question please leave it blank.

Thank you.

Section 1 General Information

- 1.1 At the start of the project, what was the target date estimated for the initial delivery of the software to end users? _____/_____/_____
day month year
- 1.2 When was the software actually delivered to end users? _____/_____/_____
day month year
- 1.3 What was the total cost of the development project? \$ _____
- 1.4 How many person-months were dedicated to the development project? _____
- 1.5 What proportion of the project's total person-months (your answer to 1.4) was invested
(a) between key points A and B? _____% (b) between key points B and C? _____%
(c) between key points C and D? _____% (d) between key points D and E? _____%
- 1.6 How many individuals were members of the development project?
Full-time _____ Part-time _____
- 1.7 How many engineers or computer scientists were members of the development project?
Full-time _____ Part-time _____
- 1.8 How many individuals developed and assessed prototypes during the project?
Full-time _____ Part-time _____
- 1.9 How many of the individuals who developed and assessed prototypes during the project were engineers and computer scientists?
Full-time _____ Part-time _____
- 1.10 How many engineers and computer scientists were members of both the prototyping team and the product development team?
Full-time _____ Part-time _____
- 1.11 What was the number of new source lines of code developed? _____
- 1.12 What was the number of reused source lines of code? _____
- 1.13 How satisfied were you that the software met customer requirements?
Very Dissatisfied					Very Satisfied
▽					▽
1		2	3	4	5

APPENDIX C

Section 2 Prototype Development

- 2.1 How many prototypes were constructed during the project? _____
- 2.2 How many person-months were dedicated to the development and assessment of prototypes? _____
- 2.3 What proportion of the total person-months dedicated to the development and assessment of prototypes (your answer to 2.2) was invested
- (a) between key points A and B _____% (b) between key points B and C _____%
- (c) between key points C and D _____% (d) between key points D and E _____%
- 2.4 How many customer sites were involved in the testing of prototypes between key points D and E? _____
- 2.5 Indicate with an "X" whether one or more of the personnel referred to in the columns actively participated in meetings and design activities that affected prototype development

	Customer employees	Supplier employees	Senior managers of the company
between key points A and B			
between key points B and C			
between key points C and D			
between key points D and E			

- 2.6 Indicate with an "X" whether one or more members of the company's functions shown in the columns actively participated in meetings and design activities that affected prototype development

	Marketing & Sales	Product Management	Software Design	Customer Service	Quality	Software Maintenance
between key points A and B						
between key points B and C						
between key points C and D						
between key points D and E						

- 2.7 What approaches were used to construct the prototypes?
- (a) throwaway _____ (b) evolutionary _____ (c) mixed _____ (d) other _____
- _____ (please specify)

APPENDIX C

2.8 The functionality of the end product embodied by the prototypes at:

- (a) key point A was _____ % (b) key point B was _____ %
(c) key point C was _____ % (d) key point D was _____ %

2.9 The real-time performance of the end product embodied by the prototypes at:

- (a) key point A was _____ % (b) key point B was _____ %
(c) key point C was _____ % (d) key point D was _____ %

2.10 The ease of use of the end product embodied by the prototypes at:

- (a) key point A was _____ % (b) key point B was _____ %
(c) key point C was _____ % (d) key point D was _____ %

2.11 The quality (overall, the software meets customer requirements and rarely breaks down) of the end product embodied by the prototypes at:

- (a) key point A was _____ % (b) key point B was _____ %
(c) key point C was _____ % (d) key point D was _____ %

Section 3 Tools and Languages

3.1 What were the top three tool sets used to develop the prototypes and when were they first used in the organization?

	Toolset Name	First used (month/year)
1		
2		
3		

3.2 What were the top three simulation and analysis languages used to develop the prototypes and when were these languages first used in the organization?

	Language	First used (month/year)
1		
2		
3		

Section 4 Project Outcomes

4.1 For the xxxx project, what were the most important prototype management decisions that contributed to shortening software project duration?

4.2 For the xxxx project, what were the most important prototype management decisions that contributed to reducing schedule overrun?

4.3 For the xxxx project, what were the most important prototype management decisions that increased your satisfaction that the software met customer's needs?

In the case of missing responses, I may need to contact you personally. Be assured that your responses will be held in strictest confidence. Please provide your:

Name: _____ E-mail: _____

Phone Number: (____) _____

Please check here ☐ if you wish to receive a copy of the research results

THANK YOU FOR YOUR TIME

APPENDIX D

Appendix D
Correlations for Independent Variables

Independent Variable	Number of prototypes	Number of engineers involved in both software and prototype development
Number of prototypes	1.00	
Number of engineers involved in both software and prototype development	.19	1.00
External involvement		
• across all stages	.14	.19
• at stage 1	.11	.21
• at stage 2	.09	.03
• at stage 3	.29	.16
• at stage 4	.00	.22
Functional involvement		
• across all stages	.12	.25
• at stage 1	.29	.22
• at stage 2	.10	.16
• at stage 3	.21	.23
• at stage 4	-.17	.25
Number of customer sites involved in the assessment of prototypes	.07	-.05
Effort dedicated to prototyping		
• across all stages (person-months)	-.11	.27
• at stage 1 (person-months)	-.15	.13
• at stage 2 (person-months)	-.12	.15
• at stage 3 (person-months)	-.01	.46***
• at stage 4 (person-months)	-.08	.15
Extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the product		
• at the end of stage 1	.03	.05
• at the end of stage 2	.00	-.10
• at the start of stage 3	.09	.13

p<.05; *p<.01

APPENDIX D

Appendix D
Correlations for Independent Variables (continued)

Independent Variable	External involvement across all stages	External involvement at stage 1	External involvement at stage 2	External involvement at stage 3	External involvement at stage 4
External involvement					
• across all stages	1.00				
• at stage 1	.87***	1.00			
• at stage 2	.79***	.59***	1.00		
• at stage 3	.83***	.62***	.56***	1.00	
• at stage 4	.82***	.64***	.47***	.56***	1.00
Functional involvement					
• across all stages	.38**	.43***	.22	.40	.21
• at stage 1	.28	.43***	.14	.29	.07
• at stage 2	.30	.33***	.36**	.33**	.00
• at stage 3	.34**	.36***	.15	.46***	.16
• at stage 4	.39**	.38***	.11	.29	.47
Number of customer sites involved in the assessment of prototypes	.02	.09	-.09	.04	.01
Effort dedicated to prototyping					
• across all stages (person-months)	-.25	-.29	-.26	-.21	-.10
• at stage 1 (person-months)	-.33**	-.34**	-.30	-.26	-.22
• at stage 2 (person-months)	-.26	.28	-.24	.22	-.15
• at stage 3 (person-months)	-.09	-.14	-.14	-.10	.07
• at stage 4 (person-months)	-.20	-.24	-.20	-.13	-.10
Extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the product					
• at the end of stage 1	.19	.22	.04	.16	.21
• at the end of stage 2	.10	.12	.11	.07	.04
• at the start of stage 3	.43	.48**	.30	.36**	.27

p<.05; *p<.01

APPENDIX D

Appendix D
Correlations for Independent Variables (continued)

Independent Variable	Functional involvement across all stages	Functional involvement at stage 1	Functional involvement at stage 2	Functional involvement at stage 3	Functional involvement at stage 4
Functional involvement					
• across all stages	1.00				
• at stage 1	.85***	1.00			
• at stage 2	.87***	.72***	1.00		
• at stage 3	.94***	.72***	.79***	1.00	
• at stage 4	.80***	.53***	.52***	.71***	1.00

p<.05; *p<.01

APPENDIX D

Appendix D
Correlations for Independent Variables (continued)

Independent Variable	Number of customer sites involved in the assessment of prototypes	Effort dedicated to prototyping across all stages (person-months)	Effort dedicated to prototyping at stage 1 (person-months)	Effort dedicated to prototyping at stage 2 (person-months)	Effort dedicated to prototyping at stage 3 (person-months)	Effort dedicated to prototyping at stage 4 (person-months)
Number of customer sites involved in the assessment of prototypes	1.00					
Effort dedicated to prototyping						
• across all stages (person-months)	-.10	1.00				
• at stage 1 (person-months)	-.07	.84***	1.00			
• at stage 2 (person-months)	-.18	.86***	.66***	1.00		
• at stage 3 (person-months)	-.04	.82***	.62***	.50***	1.00	
• at stage 4 (person-months)	-.08	.84***	.57***	.77***	.53***	1.00

p<.05; *p<.01

APPENDIX D

**Appendix D
Correlations for Independent Variables (continued)**

Independent Variable	Extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the product at the end of stage 1	Extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the product at the end of stage 2	Extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the product at the start of stage 3
Extent to which the prototypes embodied the functionality, real time performance, ease of use and quality of the product •at the end of stage 1	1.00		
•at the end of stage 2	.65***	1.00	
•at the start of stage 3	.46***	.55***	1.00

p<.05; *p<.01

END

2 4 - 0 | 6 - 9 | 7

FIN