A CONTINGENCY MODEL FOR SELECTING AN INFORMATION SYSTEM PROTOTYPING STRATEGY

By

BILLY CHARLES HARDGRAVE

Bachelor of Science
Arkansas Tech University
Russellville, Arkansas
1987

Master of Business Administration
Southwest Missouri State University
Springfield, Missouri
1990

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 1993

OKLAHOMA STATE UNIVERSITY

A CONTINGENCY MODEL FOR SELECTING
AN INFORMATION SYSTEM
PROTOTYPING STRATEGY

Thesis Approved:

Marilyn G. Kletke
Thesis Adviser

Kenneth R. Eastman

Patrick B. Dorr

Thomas C. Collins
Dean of the Graduate College

ACKNOWLEDGEMENTS

I must begin by expressing my appreciation to Reed Doke and Neil Swanson of Southwest Missouri State whose encouragement and support persuaded me to enter a doctoral program. A special thanks also goes to Tom Stone, Wayne Meinhart, and the OSU Graduate College for allowing me to enter the doctoral program at Oklahoma State.

An obvious debt of gratitude is owed to my dissertation committee - Ken Eastman, Meg Kletke, Nik Dalal, Pat Dorr - it has been a pleasure to work with them. We often hear the 'horror' stories associated with the dissertation committees. However, in my case, these stories are unfounded. I have enjoyed working with each and every one of them. Their efforts and support are greatly appreciated.

A special thanks goes to my advisor, Rick Wilson. Dr. Wilson is much more than an advisor - he is a friend, a mentor, and a colleague. I have enjoyed working with Dr. Wilson, and, to this point, our joint efforts have been successful. I would have never made it through the Ph.D. program without Dr. Wilson's help. Thank you, thank you, thank you.

A tremendous debt of gratitude is also owed to my office-mate and friend, Kent Walstrom. We spent countless hours complaining, having fun, and (productively) working together. The 'Bent NAlgorithm' will forever live as the solution to the doctoral student/faculty relationship question. Kent helped me (in his own way) get by.

I must also thank my family, friends, and in-laws for their support. A special "Thank you" goes to Cecil and Thelma Grier for their extraordinary support.

I dedicate this dissertation to my wife, Ronda. It is amazing that Ronda has tolerated so much these past three years. Anybody else would have deserted me a long time ago. This dissertation is as much hers as it is mine - I could not have done it without her.

I also dedicate this dissertation to my dad, the late Glen N. Hardgrave, Sr. I wish he were here to see me finish. He never pushed, but always encouraged. On many occasions when I thought of quitting, thoughts of him kept me going. I know he has been watching my progress - I hope I have made him proud.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF TABLES

Table | Page

Table Page

LIST OF FIGURES

Figure | Page

CHAPTER I

INTRODUCTION

1.1 Software Crisis

Improving the quality of software development has been identified as an important issue by practitioners and academicians. By practitioners, the issue has been consistently ranked in the top ten of the most critical issues in IS management, with the exception of Brancheau and Wetherbe's survey (1987) which ranked it 13th (Dickson, Leitheiser, Wetherbe, and Nechis, 1984; Hartog and Herbert, 1986; Niederman, Brancheau, and Wetherbe, 1991). By academicians, the issue is ranked in the top five of the most meaningful research areas (Teng and Galletta, 1990).

Evidence of the importance of improving the development process appears in the literature. One study indicated a 12 percent annual growth in demand for software applications, but only a four percent annual software productivity growth rate (Martin, 1983). One result of this phenomenon is a three to four-year application backlog. If demand continues to outpace supply, the backlog for applications will maintain its growth (Swanson and Beath, 1990).

Additionally, many of the systems developed today exceed budget and time constraints, and do not meet user expectations. Systems that experience one or more of these factors can be classified as failures (Ives and Olson, 1984; McKeen, 1983; Saarinen, 1990). Lyytinen's (1988) study estimates that 20 to 50 percent of all information systems fail.

The aforementioned are only a few of the problems contributing to a condition described in the software industry as a "software crisis" (Pressman, 1992). Numerous tools, techniques, and methodologies have been developed and promoted in an effort to improve systems development and eliminate the "software crisis." Recently, prototyping has emerged as a promising solution.

Many studies have shown that prototyping can improve the quality of systems (e.g., Alavi, 1984a; Alter and Ginzberg, 1978; Berrisford and Wetherbe, 1979; Kraushaar and Shirland, 1985; Mahmood, 1987). Prototyping is an approach that can be used to reduce the applications backlog by producing systems more quickly and effectively than the traditional systems development approach (Kraushaar and Shirland, 1985). However, there is disagreement in the literature over what prototyping is, how it should be used, and under what conditions it should be used.

1.2 The Prototyping Approach

The term "information systems prototype" has no unique definition (Alavi, 1984a; Sroka and Rader, 1986). Instead, several definitions and related typologies of prototypes exist. The next section will identify some of the most common typologies and provide a definition of prototyping derived from the typologies.

1.2.1 What is Prototyping?

Several typologies, used to define prototyping, exist. Doke (1990) identifies the following four types of prototypes:

1. Illustrative: Produces only mockups of reports and screens. Prototype is discarded after use.
2. Simulated: Simulates some system functions, but does not use real data. Prototype is discarded after use.
3. Functional: Performs some actual system functions and uses real data. Prototype is discarded after use.
4. Evolutionary: Produces prototype(s) that become part of the final operational system. Prototype is retained and utilized.

Cerveny, Garrity, and Sanders (1986) provide the following three category typology:

1. I/O design: Provides mockups of printed reports and/or on-line screens. Prototype is discarded after use.
2. Heuristic design: Prototype includes limited interaction of files and

transactions. Prototype is discarded after use.

3. Adaptive design: The user and builder experiment with the prototype until an effective and complete system is produced. Prototype is retained and utilized.

Huffaker (1986) identifies two types of prototypes:

1. Expendable: Prototypes are discarded when they are no longer needed. They do not become part of the final operational system.
2. Evolutionary: Prototypes evolve into operational systems. Prototype is retained and utilized.

Other typologies are provided by Carey (1990), Carey and Mason (1983), Graham (1989), Gronbaek (1989), Klingler (1988), Sethi and Teng (1988), and Slusky (1987).

What do all of these typologies have in common? Prototyping is seen as a model of the final system. The final system is either built from scratch and the prototype is discarded, or the final system is evolved from the prototype. Because of this commonality, the expendable/evolutionary typology will be used in this study to classify prototypes. In two categories, this typology captures the attributes of the other typologies. Therefore, the following definition for "prototype" will be used in this study:

> An information system prototype is a model of a system. A prototype can be as simple as mock-ups of reports or screens or as complete as software that actually does some processing. Prototypes can be built with the intention of discarding them after they are no longer needed (expendable prototype) or they become part of the final operational system (evolutionary prototype). Prototyping is the process of developing prototypes.

During the course of this study, when a distinction between the types of prototypes is not necessary, the terms "prototype" and "prototyping" will be used to generically refer to either an expendable or evolutionary prototype. Also, the term "information system," as used herein, indicates both transaction processing support systems (TPS) and information reporting systems (IRS), but excludes decision support systems (DSS). A TPS processes routine transactions in a cost efficient manner (Zmud, 1983). An IRS focuses on directing attention, providing clues, and reviewing past performance (Zmud, 1983). A DSS supports ill-structured and situation-specific decision making activities (Zmud, 1983).

1.2.2 How Should Prototyping be Used?

Conflicting reports of how prototyping should be used appear in the literature. In identifying how prototyping should be used, prototyping is viewed in the context of its relationship to the system development life cycle (SDLC).

Prototyping is commonly viewed as: (1) a replacement for the SDLC; (2) a replacement for various phases of the SDLC; and (3) a tool to improve the SDLC.

1.2.2.1 An Overview of the Systems Development Life Cycle. The most commonly used development methodology is the system development life cycle (Necco, Gordon, and Tsai, 1987). This methodology is based on a series of linear steps which, if followed, should ensure the development of a satisfactory system (Dennis, Burns, and Gallupe, 1987; Gavurin, 1991; Weinberg, 1991). Although many variations exist, typical SDLCs consist of the following steps: Analysis, Design, Coding, and Implementation (Dennis, Burns, and Gallupe, 1987).

The main objective of the SDLC is to strictly control the development process via an adherence to the SDLC phases and specific documentation. The documentation serves as a "contract" between the user and the developer. It is often mandatory that the user "sign-off" that the specifications (i.e., documentation) are correct and accurate before the project is to proceed (Gavurin, 1991; Swift, 1989). Once the user has "signed-off," changes are not permitted until the system is installed (Dennis, Burns, and Gallupe, 1987).

A benefit of this approach is its strict establishment of controls and its structured approach to systems development (Gavurin, 1991). This facilitates the development of a system within the project's functional and budgetary constraints.

However, the SDLC has been criticized because of its strict linearity. Most important, requirements missed in the beginning of the project may be recognized too late to include in the system (Dennis, Burns, and Gallupe, 1987), resulting in a system that does not meet the user's needs. Also, poor communication between developers and users during the analysis phase causes the majority of errors and are the most expensive to correct (Sroka and Rader, 1986; Slusky, 1987).

1.2.2.2 Prototyping as a Replacement to the SDLC. Prototyping is viewed by some as a replacement, or alternative, to the SDLC (Naumann and Jenkins, 1982; Swift, 1989; Tozer, 1987). As a replacement to the SDLC, prototyping becomes a software development methodology. Prototyping, as a methodology, is the same as the "evolutionary prototype" identified earlier.

Naumann and Jenkins (1982) identify the prototyping methodology as a four step procedure:

1. Identify basic requirements: identify the essential features; completeness is not important.
2. Develop working prototype: this must be accomplished very quickly (e.g., an "overnight" development of a prototype).
3. Implement and use: hands-on use of the system provides experience, understanding, and evaluation.
4. Revise and enhance: undesirable or missing features identified by the user must be corrected.

NOTE: The last two phases (3 and 4) are repeated until the system is completed.

The distinguishing feature of the view of prototyping as an alternative to the SDLC is the evolutionary nature of the prototype. Essentially, the system is built in small segments, using prototypes, until the prototype becomes the final system. Obviously, this procedure does not involve a linear sequence of phases of development, or a "sign-off" requirement from users.

1.2.2.3 Prototyping as a Replacement for SDLC Phases. Prototyping is also viewed as a replacement for one or more phases of the SDLC (Boar, 1986) - although most see prototyping as a replacement only for the analysis phase of the SDLC (Davis, 1982; Gutierrez, 1989; McKeen, Naumann, and Davis, 1979; Ryckman, 1987; Sethi and Teng, 1988; Teng and Sethi, 1990). Those advocating this view contend that prototyping is the best method for extracting requirements from users. The traditional documentation that accompanies the SDLC is not necessary - the prototype can replace the analysis phase (Gutierrez, 1989; Ryckman, 1987).

According to Sethi and Teng (1988), prototyping follows decision analysis or data analysis in the analysis phase of the SDLC. The prototype is used as a quick implementation of an intentionally incomplete system. The view is one of discovering from experimentation. Later, Teng and Sethi

(1990) evaluated prototyping as an alternative to decision analysis or data analysis as an analysis technique.

Davis (1982) identified prototyping as one of four possible approaches for determining information requirements during the analysis phase. The four methods are: (1) asking; (2) deriving from an existing information system; (3) synthesis from characteristics of the utilizing system; and (4) evolutionary prototyping.

1.2.2.4 Prototyping as a Tool. Finally, prototyping may also be considered as a tool to enhance, or support, the SDLC (Adamski, 1985; Cerveny, Garrity, and Sanders, 1986; Cerveny, Garrity, Hunt, Kirs, Sanders, and Sipior, 1987; Dearnley and Mayhew, 1983; Dennis, Burns, and Gallupe, 1987; Doke, Hardgrave, and Swanson, 1991; Gavurin, 1991; Hardgrave, Doke, and Swanson, 1993; Harrison, 1985; Necco, Gordon, and Tsai, 1987; Weinberg, 1991). This view advocates that prototyping can be used in conjunction with the traditional SDLC to augment the users' participation in, and understanding of, the requirements, and conceptual and detailed design stages (Harrison, 1985). Prototyping, as a tool, is similar to the view expressed in the previous section (prototyping as a replacement for SDLC phases). The only difference is that prototyping is used to enhance or support the SDLC, but not replace it. The traditional documentation of the SDLC is not neglected because of the prototype.

To determine how prototyping was actually used in industry, Hardgrave, Doke, and Swanson (1993) tested the relationship between prototyping and each phase of the SDLC, as used by Fortune 1000 companies. Results of their study indicated that prototyping was primarily used as a tool to support, or enhance, the SDLC. Ninety-two percent of the respondents to their survey indicated that prototyping was used to enhance the analysis phase; 83 percent used prototyping to support the design phase; and 70 percent used it during the coding phase. Only 14 percent replaced the analysis phase with prototyping; 17 percent replaced the design phase with prototyping; and 20 percent used prototyping to replace the coding phase of the SDLC. The conclusion from this study was that prototyping was used, in practice, as a tool to enhance the SDLC.

1.2.2.5 A Summary of the Prototyping Approach. As shown, two basic types of prototypes exist: expendable and evolutionary prototypes. Although more refined typologies exist, they all fit within the two-category classification. Also, prototyping is viewed as a method to replace the SDLC, or as a tool to enhance the SDLC. Prototyping is also viewed as a way to replace SDLC phases; although empirical evidence does not support this view (Hardgrave, Doke, and Swanson, 1993).

The advantages gained from the proper employment of prototypes can be summarized as: (1) systems can be developed much faster (Carey, 1990; Berrisford and Wetherbe, 1979); (2) systems are easier for end-users to learn and use (Carey, 1990); (3) development backlogs can be decreased (Carey, 1990); (4) prototyping facilitates end-user involvement (Alavi, 1984a; Berrisford and Wetherbe, 1979; Carey, 1990); (5) system implementation is easier because users know what to expect (Carey, 1990); (6) user requests are easier to determine (Carey, 1990); (7) development costs are reduced (Carey, 1990); (8) the resultant system is the right system and needs little changing (Alavi, 1984a; Carey, 1990); and (9) greater user satisfaction (Berrisford and Wetherbe, 1979).

However, prototyping does present some risks. The following are the more common risks associated with improperly using prototypes: (1) inappropriate, incomplete, and inadequate analysis and design (Carey, 1990; Weinberg, 1991); (2) unrealistic performance expectations (Alavi, 1984a; Carey, 1990; Weinberg, 1991); (3) poorly controlled projects (Alavi, 1984a; Gupta, 1988; Weinberg, 1991); (4) reluctance to discard expendable prototypes (Berrisford and Wetherbe, 1979; Carey, 1990; Weinberg, 1991); (5) problems with users (Alavi, 1984a; Weinberg, 1991); (6) lack of documentation (Gupta, 1988); (7) lack of efficiency of the system when using evolutionary prototypes (Carey, 1990; Gupta, 1988); and (8) prototyping may require specialized tools (Alavi, 1984a).

1.3 Purpose of This Study

As mentioned in Section 1.1, there is disagreement in the literature over what prototyping is, how it should be used, and under what conditions it should be used. The first two issues - what is prototyping (Section 1.2.1), and how should it be used (Section 1.2.2) - have been addressed earlier. The last issue - under what conditions should it be used - is the focus of this study.

When used properly, prototyping provides many advantages. However, it is a not a panacea for all problems associated with systems development. If not used properly, prototyping can be counterproductive (Gilhooley, 1987). Additionally, the type of prototype must be considered when using prototyping. It has been suggested that contingency models, which identify alternatives based upon situations, should be used to clarify the choice problem (Saarinen, 1990). By identifying characteristics surrounding a particular project, an appropriate prototyping strategy can be determined. A properly selected prototyping strategy can, in turn, increase the likelihood of system success. A strategy, as used here, is defined as a general approach for achieving an objective (Davis, 1982; Naumann, Davis, and McKeen, 1980).

The primary purpose of this study is to gather evidence which will indicate the characteristics influencing the selection of a prototyping strategy. This study is exploratory in the sense that little previous research has been conducted in the area of prototyping strategy selection, and no known theory of prototyping strategy selection exists. As an exploratory study, three sub-purposes exist (Kerlinger, 1973): (1) to discover significant variables affecting the prototyping strategy decision; (2) to discover relationships among variables; and (3) to lay the groundwork for later, more systematic and rigorous testing of hypotheses. Based upon the results of this study, a contingency model for selecting a prototyping strategy will be proposed.

1.3.1 Significance of the Study

Many tools, techniques, and methodologies have been developed in response to the "software crisis." Unfortunately, many of these have not proven to be effective. Prototyping is an exception. It has found wide support and continues to grow in interest.

Recent studies have provided evidence of the acceptance of prototyping. As reported by Langle, Leitheiser, and Naumann (1984), 33 percent of the respondents were using prototyping and 21 percent were considering their use. A few years later, a survey by Necco, Gordon, and Tsai (1987) found that 46 percent were using prototyping and 29 percent were considering using prototyping. That same year, a survey by Schultz and Eierman (1987) indicated a 34 percent usage rate. Carey and McLeod's (1988) study indicated that 49 percent of the respondents used prototyping. Separate studies by Doke (1990) and Saarinen (1990) found usage rates of 61 percent and 33 percent, respectively. The results of the above surveys indicate that prototyping is growing in acceptance, but is not used by **all** companies. In fact, only one study (Doke, 1990) found a usage rate of more than 50 percent.

The fact that prototyping is not used by all companies reinforces the claim that prototyping is not a panacea for all software development projects (Gilhooley, 1987; Klingler, 1988). According to Iivari and Koskela (1987):

> The proliferation of IS design methodologies, methods, techniques, and tools implies a problem of selecting appropriate methodologies, etc., for each situation. The contingency idea, that there is no detailed IS design methodology which is best in all situations, is widely accepted now.

The advantages provided earlier can only be realized if the proper prototyping strategy is employed. The improper use of prototyping can result in various risks, as summarized previously. Thus, the selection of an appropriate prototyping strategy, based upon various characteristics, is important.

This study represents the first attempt to build a comprehensive model suggesting the conditions for which prototyping is beneficial. The resulting (proposed) contingency model provided by this study is important for both practitioners and academicians. For practitioners hesitant to use prototyping, it provides a mechanism to assist the adoption of prototyping by identifying those projects that would be most benefitted by prototyping. For practitioners currently using prototyping, the model can provide direction toward the proper use of prototyping. For academicians, the model can provide directions for future research. By empirically investigating the conditions under which prototyping is beneficial, research in the area of prototyping can become more focused and cohesive.

1.4 Organization of Research

This dissertation is organized into six chapters as follows. Chapter I has provided a basic overview and background on the importance of this study.

Chapter II reviews the relevant literature. This review provides a discussion of the previous research in the area of prototyping selection. A brief discussion of system success is also provided. The chapter concludes with an overview of contingency theory and a discussion of the use of contingency models in information systems development research.

Chapter III presents the propositions and hypotheses developed in this study, and their derivation. The subjects, data collection method, and research methodology used to test these hypotheses, are described. A preliminary discussion of data analysis methods is also provided.

Chapter IV presents the results of the data collection. A discussion of sample representativeness and instrument reliability and validity are provided. The primary focus of the chapter is to provide the results of the tests of Hypotheses 1 through 19.

Chapter V discusses the results presented in Chapter IV. The data analysis used to validate the contingency model is also provided. The final part of the chapter is used to propose a contingency model of prototyping strategy selection.

Chapter VI summarizes the findings of the study, examines the limitations of the study, and discusses directions for future research.

CHAPTER II

REVIEW OF RELEVANT LITERATURE

2.1 Overview

This chapter examines the literature pertinent to this study. First, the prototyping literature that is relevant to the identification of factors used for selecting a prototyping strategy is examined. Each of these factors will, in turn, be discussed. Second, a brief review of the impact of prototyping on system success, and traditional definitions of system success, will be presented. Third, an overview of contingency theory, and the use of contingency models in software development is presented. Finally, a summary of the chapter is presented.

2.2 Factors Affecting Prototyping Strategy Selection

Much of the existing evidence of prototyping strategy selection is qualitative, based on individual cases, conceptual discussions, and technical literature (Iivari and Karjalainen, 1989; Mahmood, 1987; Teng and Sethi, 1990). Data is seldom derived from an empirical basis, and often amounts to little more than "armchair speculation" (McKeen, 1983). Unfortunately, speculation often becomes a factor in strategy selection without empirical validation. This information tends to confuse, rather than clarify, the proper uses of prototyping.

As presented in the following sections, many factors influencing the selection of a prototyping strategy are considered. The factors have been classified as: (1) project characteristics; (2) user characteristics; (3) developer characteristics; and (4) organization characteristics. The factor groupings were made based upon logical association. At this time, the four groups of characteristics are nothing more than a framework for discussing the factors.

The factors are summarized in Tables 1 through 4. For each of the factors identified, the contingency's meaning and derivation (i.e., conceptually derived, case study, survey, field study, etc.) will be provided. Also, note the number of factors and contradictory suggestions.

2.2.1 Project Characteristics

Project characteristics represent the largest set of factors affecting the selection of a prototyping strategy. The major characteristics, discussed in detail in the following sections and summarized in Table 1, include: clarity of requirements, requirements stability, system mode, project duration, innovation, project size, project impact, and performance.

2.2.1.1 Clarity of Requirements. The clarity of system requirements is often cited as influencing the prototyping decision. The traditional SDLC is designed to create a complete and correct set of requirements before the system is designed and built. If the requirements cannot be determined correctly and completely, the system is rejected by users, or must receive substantial rework to fit users' needs. Prototyping captures an initial set of requirements, and through iterative discovery, builds the system to meet the users' needs. Thus, when system requirements are unclear, or users are vague or ambiguous, a prototyping approach can help clarify the needs of the users and requirements of the system (Alavi, 1984a; Asner and King, 1981; Berrisford and Wetherbe, 1979; Burns and Dennis, 1985; Carey and Currey, 1989; Connell and Brice, 1985; Davis, 1982; Dos Santos, 1986, 1988; Gavurin, 1991; Gremillion and Pyburn, 1983; Janson and Smith, 1985; Kauber, 1985; Kraushaar and Shirland, 1985; Mahmood, 1987; Naumann and Jenkins, 1982; Naumann, Davis, and McKeen, 1980; Slusky, 1987; Smith, 1987; Yaverbaum, 1989). Prototyping can help identify the problem as well as solve it (Gremillion and Pyburn, 1983). However, when requirements are very clear, projects are easy to manage and produce. Prototyping may increase development time when specifications are unambiguous by involving users more than necessary (Gavurin, 1991).

A Delphi study by Doke, Swanson, and Hardgrave (1992) found "unclear requirements" to be the number one reason for using prototyping. Respondents to Carey and Currey's (1989) survey indicated "unclear requirements" as the number two reason. Other studies have also indicated the existence of unclear requirements as a reason to use a prototyping approach (Alavi, 1984a; Guimaraes, 1981). However, a survey by Saarinen (1990) indicated that, in practice, this contingency was not followed. Sixteen of 23 projects reported in Saarinen's (1990) study did not use a prototyping approach in the face of unclear requirements.

Prototyping when a project exhibits unclear requirements is not universally accepted. Carey (1990) argues that prototyping should be used only for well-defined problems because of the difficulty in managing the project. Krzanik (1986) recommends using an expendable prototype for projects with clear requirements, and an evolutionary prototype for unclear requirements. Lynch (1987) suggests that prototyping can be used for both well- and ill-defined projects.

2.2.1.2 Requirements Stability. Requirements stability refers to the stability of the project requirements during development. Stability can be affected by two sources: users and the organization. Users cannot always fully specify requirements at the beginning of the project. In this case, requirements may change during the development of the system. When users cannot specify requirements fully, prototyping can provide a base from which adjustments can be made (Davis, 1982).

The organization may also affect project stability because of the business environment. If the environment is volatile, it may affect project requirements (Dos Santos, 1988). In a volatile business environment, detailed specification decisions must be delayed as long as possible. Prototyping facilitates this due to its incremental nature (Dos Santos, 1988).

In general, researchers agree that prototyping is the best strategy when unstable (i.e., dynamic) project requirements exist (Klingler, 1986, 1988; Kraushaar and Shirland, 1985; Li, 1990; Smith, 1987). A prototype, since it is being developed iteratively, readily accommodates changes. Using a "no-prototyping" strategy, a change in requirements may not be possible until the system is completely finished (Davis, 1982; Klingler, 1986; Kraushaar and Shirland, 1985; Naumann and Jenkins, 1982).

There are a few who do not agree with the view that any type of prototype is appropriate in a dynamic environment. Carey (1990) and Krzanik (1986) recommend only an evolutionary prototyping strategy for a dynamic environment. For stable environments, Krzanik (1986) suggests an expendable prototype. Neither Carey (1990) or Krzanik (1986) provide explanations for their views.

2.2.1.3 System Mode. System mode, categorized as either on-line or batch, has been identified as a factor in selecting a prototyping strategy. On-line systems, also referred to as interactive systems, require a greater concentration on user interfaces than batch systems. Because of this, on-line systems are much more related to user needs and expectations than batch systems, and a prototyping approach should be employed (Burns and Dennis, 1985; Carey, 1990; Carey and Currey, 1989; Gavurin, 1991; Graham, 1989; Klingler, 1986, 1988; Mahmood, 1987; Mason and Carey, 1983; Necco, Gordon, and Tsai, 1987; Smith 1987).

A survey by Carey and Currey (1989) indicated that all prototyping efforts involved on-line systems. A different survey by Necco, Gordon, and Tsai (1987) indicated that 98 percent (of those utilizing prototyping) used prototyping for on-line systems, compared to only 42 percent for batch systems. Finally, responses of developers to Mahmood's (1987) survey indicated that prototyping should not be used for batch systems.

Prototyping allows users to participate with the interface required of an on-line system. Batch systems usually do not require an interface. However, if an interface is required, an expendable prototype can be used (Cerveny, Garrity, and Sanders, 1986).

2.2.1.4 Project Duration. Project duration, the time between the start of a project and the delivery of the final system to the user, is an important consideration in the selection of a development strategy. Long-running projects encounter many problems caused by organizational and individual changes as a direct consequence of the passage of time. The longer the duration of a project, the greater the changes are likely to be and the greater the risk that the project will become unmanageable. A project developed over six months will have to deal with fewer changes than a system developed over three years (Dos Santos, 1988).

Dos Santos (1986, 1988) provides the following delineation for duration: (1) short (under six months); (2) average (6 - 12 months); and (3) long (over 12 months). It should be noted that duration is a function of time. The size of the project can obviously affect the duration, but increasing the manpower of a project can decrease the duration and relieve the problems due to time. Likewise, a small project can be prolonged due to a shortage of manpower and suffer the problems of a long project duration.

Suggestions for choosing a prototyping strategy, based on duration, are contradictory. Dos Santos (1986, 1988) contends that an expendable prototyping strategy is best suited for a short duration project. Smith (1987) and Li (1990) also advocate using prototypes for projects with a short duration, but make no distinction for the type of prototype. Krzanik (1986) recommends using an expendable prototype for a project of average duration. Guimaraes (1981), basing his suggestions on a case study with a project duration of over six years, recommends prototyping for projects of long duration. Similarly, Graham (1989) argues that only an evolutionary prototyping strategy should be used for projects of more than six months duration.

2.2.1.5 Innovation. Innovation is an indicator of the foundation of the project; i.e., is it a new development (high innovation) or a modification to an existing system (low innovation)? Innovation, as a characteristic for selecting a prototyping strategy, provides mixed contingencies. Harrison (1985) contends that the prototyping approach can be applied to new development and modifications to systems. For modifications, Harrison (1985) suggests using the current system as a prototype. Johnson (1983) recommends, based on case study experience, the use of an expendable prototyping strategy for low innovation projects.

Others advocate using prototyping only for new development (Dos Santos, 1988). A survey by Carey and Currey (1989) indicated that prototyping was only used for new development projects. Johnson (1983) suggests that only an evolutionary prototyping strategy be used for new developments.

2.2.1.6 Project Size. Perhaps the greatest source of disagreement involving a characteristic and its influence on the selection of a prototyping strategy, is the project size. Some suggest that only large systems should use prototyping, others advocate only small systems, and some suggest that project size is not a consideration. It must be noted that in this context project size refers to the man-hours and/or cost necessary to produce the system (Burns and Dennis, 1985; Lynch, 1987; Saarinen, 1990). Project size does not include project duration or number of users, both of which are considered separately in Sections 2.2.1.4 and 2.2.2.3, respectively.

The argument for using prototyping for large systems is that, because of its size, specifications will change during the development of the system (Dearnley and Mayhew, 1983; Guimaraes, 1981; Gupta, 1988; Johnson, 1983). Prototyping readily accommodates change, thus proving its usefulness. Another argument is that prototyping should not be used for small projects because the costs of developing a prototype could not be justified (Gavurin, 1991).

A survey by Schultz and Eierman (1987) indicated that prototyping is more common in larger projects. Case studies of large systems employing prototyping have provided evidence of successful development (Brittan, 1980; Johnson, 1983). In a case study by Groner, Hopwood, Palley, and Sibley (1979), an expendable prototyping strategy was used successfully for a large system - the requirements analyses effort was approximately 37 man-years over 5 1/2 elapsed years!

Arguments against prototyping for large systems and for prototyping for small systems are also convincing. One argument is that prototyping is not useful for the development of large systems because, with prototyping, designers lack the detailed documentation that other methods provide (Yaverbaum, 1989). Another argument is that the larger the project, the more difficult it becomes to prototype the entire project, and managing the system development process becomes difficult (Burns and Dennis, 1985; Carey, 1990; Dennis, Burns, and Gallupe, 1987; Klingler, 1986, 1988; Lynch, 1987; Mahmood, 1987).

Other views of project size consideration exist. Kraushaar and Shirland (1985) demonstrated case studies of both large and small systems that successfully used a prototyping approach and concluded that project size was not an issue. It has been hypothesized that an evolutionary prototyping strategy should be used for large systems, and an expendable prototyping strategy used for small systems (Pliskin and Shoval, 1987; Saarinen, 1990; Shoval and Pliskin, 1988). However, data obtained from an industry survey indicated that size did not influence the strategy choice, thus the hypothesis was not supported (Saarinen, 1990).

2.2.1.7 Project Impact. The impact of the project on the organization is another project characteristic worthy of consideration. Critical systems - systems that operate, manage, and control the daily business activities - have a very broad and strong impact on the organization. Prototyping should be used for critical systems (Boar, 1986; Carey, 1990; Dos Santos, 1988; Gremillion and Pyburn, 1983). For critical systems it is extremely important that the system meet specifications - prototyping facilities this important requirement.

2.2.1.8 Performance. System performance is another project characteristic that has been identified. Tools used for evolutionary prototyping, such as fourth generation languages and database management systems, are typically inefficient in terms of using computer resources (Gavurin, 1991). Additionally, during the process of prototyping, the emphasis is on specifying requirements, not on the efficiency of the resulting code. Following these arguments, evolutionary prototyping should not be used for a system with a large database, or for a system with stringent performance requirements (Connell and Brice, 1984; Gavurin, 1991). Evolutionary prototyping should be used only for systems with a low volume of file/transaction processing or where the system is not regularly used (Andrews, 1983; Sroka and Rader, 1986).

2.2.1.9 Other Characteristics. This section discusses project characteristics that have not received wide recognition in the literature. Inclusion in this section indicates that the factors will not be considered in this study; it does not imply unimportance of the factors.

Functionality (i.e., system type) is cited as an important project characteristic (Klingler, 1988; Lipp, 1984). Cerveny, Garrity, and Sanders (1986) specify the proper match between prototyping and functionality by specifying an expendable prototype for transaction processing systems and information reporting systems, and evolutionary prototypes for decision support systems. Naumann and Jenkins (1982) maintain that the most promising candidates for prototyping are systems involving managerial functions such as planning, directing, controlling, problem solving, and decision making. Since only traditional information systems (i.e., TPS, IRS, etc.) are considered in this study, functionality will not be considered as a factor in prototyping strategy selection.

Other factors that have been identified include: (1) the project requires quick delivery (Asner and King, 1981; Smith, 1987); and (2) user and developer difficulties are expected (Alavi, 1984a; Berrisford and Wetherbe, 1979; Naumann and Jenkins, 1982). For the first factor identified, prototyping increases development speed, and can get something to the users quickly. For the second factor, prototyping increases communication and interaction between the user and developer, hopefully easing tension between them.

2.2.2 User Characteristics

Several user characteristics are identified as important factors in selecting a prototyping strategy. Attributes identified include user contribution, experience with prototyping, number of users, and impact of the system on the user. The characteristics are summarized in Table 2.

TABLE 1

FACTORS AFFECTING PROTOTYPING STRATEGY SELECTION:
PROJECT CHARACTERISTICS

Characteristic	Researcher(s)	Type of Study	Contingency (to use prototyping)
Clarity of Requirements	Alavi, 1984a	Field Study	Unclear requirements
	Asner and King, 1981	Case Study	Unclear requirements
	Berrisford and Wetherbe, 1979	Case Study	Unclear requirements
	Burns and Dennis, 1985	Conceptual	Unclear requirements
	Carey, 1990	Conceptual	Clear requirements
	Carey and Currey, 1989	Survey	Unclear requirements
	Connell and Brice, 1985	Conceptual	Unclear requirements
	Davis, 1982	Conceptual	Unclear requirements
	Doke, Swanson, Hardgrave,1992	Field Study	Unclear requirements
	Dos Santos, 1986, 1988	Conceptual	Unclear requirements
	Gavurin, 1991	Conceptual	Unclear requirements
	Gremillion and Pyburn, 1983	Conceptual	Unclear requirements
	Guimaraes, 1981	Conceptual	Unclear requirements
	Janson and Smith, 1985	Case Study	Unclear requirements
	Kauber, 1985	Conceptual	Unclear requirements
	Kraushaar and Shirland, 1985	Case Study	Unclear requirements
	Krzanik, 1986	Conceptual	Evolutionary prototyping for unclear requirements
	Krzanik, 1986	Conceptual	Expendable prototyping for clear requirements
	Lynch, 1987	Conceptual	Clear or unclear requirements
	Mahmood, 1987	Conceptual	Unclear requirements
	Naumann and Jenkins, 1982	Conceptual	Unclear requirements
	Naumann, Davis, McKeen, 1980	Case Study	Unclear requirements
	Saarinen, 1990	Survey	Prototyping not needed for unclear requirements
	Slusky, 1987	Conceptual	Unclear requirements
	Smith, 1987	Conceptual	Unclear requirements
	Yaverbaum, 1989	Conceptual	Unclear requirements

TABLE 1 (continued)

Characteristic	Researcher(s)	Type of Study	Contingency (to use prototyping)
Requirements Stability	Carey, 1990	Conceptual	Evolutionary prototyping for unstable requirements
	Davis, 1982	Conceptual	Unstable requirements
	Dos Santos, 1988	Conceptual	Unstable requirements
	Klingler, 1986, 1988	Conceptual	Unstable requirements
	Kraushaar and Shirland, 1985	Case Study	Unstable requirements
	Krzanik, 1986	Conceptual	Evolutionary prototyping for unstable requirements
	Krzanik, 1986	Conceptual	Expendable prototyping for stable requirements
	Li, 1990	Conceptual	Unstable requirements
	Smith, 1987	Conceptual	Unstable requirements
System Mode	Burns and Dennis, 1985	Conceptual	On-line system
	Carey, 1990	Conceptual	On-line system
	Carey and Currey, 1989	Survey	On-line system
	Cerveny, Garrity,Sanders,1986	Conceptual	Expendable prototyping for batch system with interface
	Gavurin, 1991	Conceptual	On-line system
	Graham, 1989	Conceptual	On-line system
	Klingler, 1986, 1988	Conceptual	On-line system
	Mahmood, 1987	Survey	On-line system
	Mason and Carey, 1983	Conceptual	On-line system
	Necco, Gordon, Tsai, 1987	Survey	On-line system
	Smith, 1987	Conceptual	On-line system
Project Duration	Dos Santos, 1986, 1988	Conceptual	Expendable prototyping for short duration
	Graham, 1989	Conceptual	Evolutionary prototyping for long duration
	Guimaraes, 1981	Conceptual	Long duration
	Krzanik, 1986	Conceptual	Expendable prototyping for average duration
	Li, 1990	Conceptual	Short duration
	Smith, 1987	Conceptual	Short duration

TABLE 1 (continued)

Characteristic	Researcher(s)	Type of Study	Contingency (to use prototyping)
Innovation	Carey and Currey, 1989	Survey	High innovation
	Dos Santos, 1988	Conceptual	High innovation
	Harrison, 1985	Conceptual	High and low innovation
	Johnson, 1983	Case Study	Expendable prototyping for low innovation
	Johnson, 1983	Case Study	Evolutionary prototyping for high innovation
Project Size	Brittan, 1990	Case Study	Large system
	Burns and Dennis, 1985	Conceptual	Small system
	Carey, 1990	Conceptual	Small system
	Dearnley and Mayhew, 1983	Conceptual	Large system
	Dennis, Burns, Gallupe, 1987	Case Study	Small system
	Gavurin, 1991	Conceptual	Large system
	Groner, Hopwood, Palley, and Sibley, 1979	Case Study	Expendable prototyping for large system
	Guimaraes, 1981	Conceptual	Large system
	Gupta, 1988	Conceptual	Large system
	Johnson, 1983	Case Study	Large system
	Klingler, 1986, 1988	Conceptual	Small system
	Kraushaar and Shirland, 1985	Case Study	Large and small systems
	Lynch, 1987	Conceptual	Small system
	Mahmood, 1987	Conceptual	Small system
	Pliskin and Shoval, 1987	Conceptual	Evolutionary prototyping for large systems
	Pliskin and Shoval, 1987	Conceptual	Expendable prototyping for small systems
	Saarinen, 1990	Survey	Large and small systems
	Schultz and Eierman, 1987	Survey	Large system
	Shoval and Pliskin, 1988	Conceptual	Evolutionary prototyping for large systems
	Shoval and Pliskin, 1988	Conceptual	Expendable prototyping for small systems
	Yaverbaum, 1989	Conceptual	Small system

TABLE 1 (continued)

Characteristic	Researcher(s)	Type of Study	Contingency (to use prototyping)
Project Impact	Boar, 1986	Conceptual	Critical system
	Carey, 1990	Conceptual	Critical system
	Dos Santos, 1988	Conceptual	Critical system
	Gremillion and Pyburn, 1983	Conceptual	Critical system
Performance	Andrews, 1983	Conceptual	Not evolutionary prototyping for high volume system or regularly used system
	Connell and Brice, 1984	Conceptual	Not evolutionary prototyping for large DB or stringent performance requirements
	Gavurin, 1991	Conceptual	Not evolutionary prototyping for large DB or stringent performance requirements
	Sroka and Rader, 1986	Conceptual	Not evolutionary prototyping for high volume system or regularly used system

2.2.2.1 User Contribution. User contribution involves the amount of time that the users are willing to give to the project (McKeen, 1983). Prototyping, compared to the traditional SDLC, requires more communication and interaction between the user and developer. Although at first enthusiastic, users must maintain their level of commitment to the project through each of the iterations required of the prototyping approach. Thus, it is obvious that prototyping should not be used if users do not have time to spend with development (Gibson and Rademacher, 1987; Meyer and Kovacs, 1983; Teng and Sethi, 1990).

2.2.2.2 Experience with Prototyping. Developers work closely with users during the development process. If prototyping is employed, users should have an *a priori* familiarity with the prototyping approach (Alavi, 1984a; Sroka and Rader, 1986). Users who are not familiar with prototyping often have unrealistic expectations (Alavi, 1984a; Berrisford and Wetherbe, 1979). For example, anything shown to the user in the form of a prototype may be perceived to be a fully operational system. Additionally, the viewing of prototypes may be seen as an indication of the speed at which the final system will be ready (Alavi, 1984a).

A related concept is the user's familiarity with automation (i.e., the use of computers/computerization). It is included here because if a user is unfamiliar with automation, then he/she is obviously unfamiliar with prototyping (but not vice versa). Lack of automation experience is seen as a risk inducer (Ahituv, Hadass, and Neumann, 1984). As such, prototyping is a way to decrease risk (Davis, 1982; Gavurin, 1991; Tate and Verner, 1990). Users unfamiliar with automated systems can develop a better idea of their system requirements by being exposed to a prototype (Dearnley and Mayhew, 1983; Gavurin, 1991). Tate and Verner (1990), from information obtained from a case study, contend that an evolutionary prototyping strategy can be used when users are inexperienced, provided the responses to user requests are sufficiently responsive.

2.2.2.3 Number of Users. Involving a large number of users in the evaluation and alteration of a prototype, and in generating requirements, would be difficult to coordinate. Prototyping for one or a few users is not difficult; prototyping for many is (Burns and Dennis, 1985). Modifications requested by one user must be approved by all affected users; iterations become slower and change is no longer a quick and easy task. Managing changes, and requests for changes, becomes almost impossible. Tillman (1989) refers to this as the "internal consistency nightmare." Thus, a large number of users, when involved in prototyping, can increase development time. Gavurin (1991) attempted to use small random samples of users at each iteration as a way to control the number of users, but found that it was not an effective method because new users at each iteration would provide requirements that conflicted with users of earlier iterations. Connell and Brice (1985), via a case study, concluded that the prototyping approach failed because all (many) users were allowed to evaluate the prototype at each iteration. However, Dos Santos (1988) maintains that prototyping can work well for both single (or a few) and multiple users, provided the process is managed properly.

2.2.2.4 User Impact. User impact examines the effect the project has on the way users perform their jobs. User impact is an important characteristic in the prototyping strategy decision. If the user impact is high (i.e., the project will considerably change the way users perform their jobs), there should be a high degree of interaction between the user and the developer. A high interaction will keep the user informed and reduce the likelihood of system rejection. Conversely, less interaction is needed if the project will not adversely affect the user's job (Dos Santos, 1988). Since a prototyping approach requires a large amount of interaction between the developer and the user, it would appear to be appropriate in cases of high user impact (Asner and King, 1981; Dos Santos, 1988).

Alter and Ginzberg (1978) suggest using a prototyping approach in cases where the impact cannot be predetermined. Prototyping allows the developers to determine the impact as the system is being developed and make adjustments accordingly.

TABLE 2

FACTORS AFFECTING PROTOTYPING STRATEGY SELECTION:
USER CHARACTERISTICS

Characteristic	Researcher(s)	Type of Study	Contingency (to use prototyping)
User Contribution	Gibson and Rademacher, 1987	Conceptual	User(s) must have time
	Meyer and Kovacs, 1983	Conceptual	User(s) must have time
	Teng and Sethi, 1987	Conceptual	User(s) must have time
Experience with Prototyping	Ahituv, Hadass, Neumann, 1984	Case Study	No automation experience
	Alavi, 1984a	Field Study	Need prototyping experience
	Davis, 1982	Conceptual	No automation experience
	Dearnley and Mayhew, 1983	Conceptual	No automation experience
	Gavurin, 1991	Conceptual	No automation experience
	Sroka and Rader, 1986	Conceptual	Need prototyping experience
	Tate and Verner, 1990	Case Study	No automation experience needed for evolutionary prototyping
Number of Users	Burns and Dennis, 1985	Conceptual	Small number of users
	Connell and Brice, 1985	Case Study	Small number of users
	Dos Santos, 1988	Conceptual	Both single and multiple users
	Gavurin, 1991	Conceptual	Small number of users
	Tillman, 1989	Conceptual	Small number of users
User Impact	Alter and Ginzberg, 1978	Field Study	Use if impact not determined
	Asner and King, 1981	Case Study	Use if high user impact
	Dos Santos, 1988	Conceptual	Use if high user impact

2.2.3 Developer Characteristics

Two characteristics, directly related to the system developer, are important in the selection of a prototyping strategy. The first is the developer's familiarity with the application domain of the system. A term used to describe the developer's application domain experience is "developer-task proficiency." Developer-task proficiency "... is not a measure of ability or potential but rather of directly applicable experience" (Naumann, Davis, and McKeen, 1980).

The second developer characteristic is the developer's experience with prototyping. We will use the term "developer-prototyping proficiency" to describe the developer's experience with prototyping. Each of the developer characteristics is discussed below. The characteristics are summarized in Table 3.

2.2.3.1 Familiarity With Application Domain.

A developer attribute that impacts the choice of a prototyping strategy is the developer's experience with the application domain, or similar applications. There are two views on this topic, each coming to the conclusion that a developer's experience with the application domain impacts the prototyping strategy decision. Each view is discussed in turn.

The first view is that if a developer has development experience with similar applications (i.e., high developer-task proficiency), then prototyping would not be appropriate

(Kraushaar and Shirland, 1985; Naumann, Davis, McKeen, 1980). The reason is that when the task is familiar to the developer, less user interaction is required to determine requirements. Unnecessary involvement of users may increase development time (Kraushaar and Shirland, 1985). Thus, using a prototyping approach in this situation could result in an increase in development time.

The second view is that if a developer lacks experience with similar applications (i.e., low developer-task proficiency), then prototyping is appropriate. In a study of 56 systems, Alter and Ginzberg (1978) determined that a designer lacking prior experience with similar systems increases system risk. They also found that prototyping can reduce the risk associated in this situation. For a situation involving low developer-task proficiency, prototyping allows for more interaction with the user which increases the degree of certainty that the developer will accurately and completely elicit and document the user's requirements (Naumann, Davis, and McKeen, 1980).

2.2.3.2 Experience With Prototyping. Another developer characteristic is the developer's experience with prototyping (i.e., developer-prototyping proficiency). Most agree that high developer-prototyping proficiency is needed to use a prototyping approach (Alavi, 1984a, 1984b; Davis, 1982; Kraushaar and Shirland, 1985; Lynch, 1987; Sroka and Rader, 1986).

The traditional SDLC used by most system developers prevents frequent changes by users. However, prototyping explicitly facilitates change. Frequent changes in user requirements may frustrate designers unless they are prepared to expect changes and view them positively (Alavi, 1984a; Sroka and Rader, 1986). A lab study conducted by Alavi (1984a) found that the designers using prototyping (other designers used the SDLC) perceived a higher degree of change in user specifications during the design process. Also, prototyping requires a greater level of interpersonal communication skills from the developers (Alavi, 1984b). Some developers will not have the communication skills necessary for a prototyping environment. Sroka and Rader (1986) suggest that developers be given training in prototyping and use prototyping for a small project first. This procedure will serve two purposes. First, it will expose the developer to prototyping. Second, it will test the developer's communication skills in a prototyping environment.

Developers should also have experience with special tools used for prototyping. Experience with the project's technology (hardware and software) affects the risk associated with the project (Davis, 1982; Lynch, 1987). Thus, lack of experience with prototyping tools increases project risk. Because evolutionary prototypes require special tools (see Section 2.2.4.3), experience is only necessary when using an evolutionary prototyping strategy.

Additionally, developers using evolutionary prototypes should be professionals in the use of required tools, in order to ensure that a quality prototype is developed to become part of the final product (Kraushaar and Shirland, 1985).

2.2.4 Organization Characteristics

Another set of factors that influences the prototyping strategy decision can be classified as organization characteristics. Included in this category are factors directly related to the organization, such as management and resources, among others. The following sections discuss the organization characteristics in more detail. The characteristics are summarized in Table 4.

2.2.4.1 Management Support. In a field study of 12 projects from six organizations, Alavi (1984a, 1984b) established management support as a critical factor in selecting a prototyping strategy. Specifically, management support and understanding are essential to the successful utilization of prototyping. In order to successfully apply prototyping, the organizational climate in terms of the established management and control procedures should accommodate the different nature and philosophy of the prototyping approach (Alavi, 1984b).

Due to the newness of prototyping as a methodology, there is a lack of knowledge for planning and controlling the project. Additionally, planning and controlling

TABLE 3

FACTORS AFFECTING PROTOTYPING STRATEGY SELECTION:
DEVELOPER CHARACTERISTICS

Characteristic	Researcher(s)	Type of Study	Contingency(to use prototyping)
Familiarity with Application Domain	Alter and Ginzberg, 1978	Field Study	No experience with similar apps
	Kraushaar and Shirland, 1985	Case Study	No experience with similar apps
	Naumann, Davis, McKeen, 1980	Conceptual	No experience with similar apps
Experience with Prototyping	Alavi, 1984a	Lab Study	Need prototyping experience
	Alavi, 1984b	Field Study	Need prototyping experience
	Davis, 1982	Conceptual	Need prototyping experience
	Kraushaar and Shirland, 1985	Case Study	Need prototyping experience
	Lynch, 1987	Conceptual	Need prototyping experience
	Sroka and Rader, 1986	Conceptual	Need prototyping experience

prototyped systems are more difficult to manage because of their evolving nature, the number of revisions, and the uncertainty in user requirements. Failure to gain management support and establish explicit procedures can lead to frustration from both users and developers, and can result in system failure (Alavi, 1984a; Voltmer, 1989).

Klingler (1988) and Voltmer (1989) agree that management support is vital to the success of a prototyped information system. Klingler (1988) recommends "selling" management on the idea of prototyping. To do this, it is necessary to present management with an explicit and well-defined life cycle that includes prototyping. Small projects can provide the opportunity to prove to management the benefits of prototyping and to illustrate the control and management support that is necessary for such projects (Klingler, 1988). Tillman (1989) also recommends selling the idea to management, and provides a five phase plan for accomplishing the task. His method is similar to Klingler's (1988), with an added emphasis that senior management, not junior management, must be involved.

2.2.4.2 Pre-Commitment. Management is often unwillingly to experiment with new concepts, ideas, and procedures, due to the risk level of such undertakings (Pressman, 1992). A "good" project may sometimes be dismissed because management will not commit resources before fully knowing the benefits. It has been suggested that prototyping be used in situations where there is a need for experimentation and learning before commitment of resources for a full project (Alavi, 1984a, 1984b; Asner and King, 1981; Mahmood, 1987; Smith 1987). The prototype provides the opportunity to test, modify, and visualize a real-life system without the resources needed for the full project. Asner and King (1981) provide the following rule of thumb: undertake the 20 percent of the work that will give 80 percent of the results. This approach will allow management to evaluate the system and make an informed decision.

The primary purpose in using prototyping in this environment is to produce something as quickly and cheaply as possible. The resulting system may have significant shortcomings in the areas of functionality, speed, and data handling (Asner and King, 1981). Thus, it would seem appropriate to use an expendable prototype.

2.2.4.3 Special Tools. The recent increase in sophisticated development tools, such as database management systems (DBMS), on-line interactive systems (e.g., fourth generation languages), application generators, text editors, word processors, and special-purpose prototyping tools, have facilitated the use of prototypes. The debate is whether these tools are necessary to employ the prototyping approach. The arguments are presented below.

A DBMS provides rapid design and programming of data handling facilities, including data input, updating, and reporting facilities (Klingler, 1988; Naumann and Jenkins, 1982). Gilhooley (1987) contends that if these elements of data management are not present, the organization must take time to implement them before adopting the prototyping approach. According to Burns and Dennis (1985), a DBMS is only necessary if using an evolutionary prototyping strategy.

An on-line interactive system (often referred to as a 4GL) is needed because the builder must respond quickly to the user's needs - batch systems do not permit interaction and revision at an acceptable rate (Alavi, 1984a; Naumann and Davis, 1982). 4GLs are capable of providing more than 10 times the productivity of third-generation programming languages (e.g., COBOL, PL/1) (Gilhooley, 1987).

Often associated with on-line interactive systems are application generators. Application generators speed the development of application programs, which facilitates the rapid response to user needs (Saarinen, 1990). An application generator would appear to be required if using an evolutionary prototyping strategy (Burns and Dennis, 1985).

Less sophisticated tools, such as text editors and word processors, allow developers to develop mock-ups of screens to demonstrate to users (Klingler, 1988). Obviously, the system has no functionality and the mock-ups are only used to demonstrate the "look" of the system (Carey and Mason, 1983).

Special-purpose prototyping tools are specifically intended to facilitate the prototyping approach to system development. For example, ACT/1 allows a developer to implement a system by entering screen examples provided by a user. The screen designs can then be transferred into the final product without extensive coding (Carey and Mason, 1983). The most popular prototyping tool, according to a survey by Carey and Mason (1983), is APL. Others include IBM's Application Development Facility (ADF) and the Chevron Program Development System (Alavi, 1984b; Carey and Mason, 1983).

Empirical evidence supporting the above arguments are inconsistent. A survey by Mahmood (1987) found that expensive support software is needed. However, it is unclear whether the survey defined the different types of prototyping. Saarinen (1990) found that of 14 projects using the prototyping approach, only six used a 4GL, the other eight used a 3GL. Again, no distinction was made between expendable and evolutionary prototyping. A case study conducted by Dennis, Burns, and Gallupe (1987) found that an evolutionary prototyping strategy was not used due to the lack of sophisticated tools. Finally, a survey by Schultz and Eierman (1987) showed that, for those companies using prototyping, 52.6 percent use 4GLs, 33.3 percent use old languages (i.e., 3GL), 12.3 percent use special-purpose prototyping tools, 19.3 percent use word processors, and 24.6 percent use spreadsheets, as prototyping tools. The logical conclusion from the conflicting reports appears to be: if an evolutionary prototyping strategy is to be used, then sophisticated tools are required; sophisticated tools are not necessary to use expendable prototypes.

2.2.4.4 Other Organization Characteristics. Other organization characteristics related to prototype selection have been investigated, but do not appear to impact the prototyping decision. These variables include organization size, age, and degree of decentralization; and MIS department variables such as size, age, and level (Alavi, 1984a; Doke, Hardgrave, Swanson, 1992; Franz and Robey, 1986; McFarlan, 1981).

2.3 Prototyping and System Success

"System success" is a hotly debated topic in the MIS literature. Since it is not our intent to provide a new definition of system success, we will adopt the following common definition for the immediate discussion: a product which arrives on schedule within budget and produces a high degree of user satisfaction is considered a successful development (Jenkins, Naumann, and Wetherbe, 1984; McKeen, 1983; Necco, Gordon, and Tsai, 1987). This definition will provide a basis for investigating evidence that prototyping increases the likelihood of system success. Much of the research concerning the use of prototypes has been posited on the belief that the **appropriate** use of prototypes has a positive influence on the success of an information system.

TABLE 4

FACTORS AFFECTING PROTOTYPING STRATEGY SELECTION:
ORGANIZATION CHARACTERISTICS

Characteristic	Researcher(s)	Type of Study	Contingency(to use prototyping)
Management Support	Alavi, 1984a	Field Study	Need management support
	Alavi, 1984b	Field Study	Need management support
	Klingler, 1988	Conceptual	Need management support
	Tillman, 1989	Conceptual	Need management support
	Voltmer, 1989	Conceptual	Need management support
Pre-Commitment	Alavi, 1984a	Field Study	If commitment evidence needed
	Alavi, 1984b	Field Study	If commitment evidence needed
	Asner and King, 1981	Conceptual	If commitment evidence needed
	Mahmood, 1987	Conceptual	If commitment evidence needed
	Smith, 1987	Conceptual	If commitment evidence needed
Special Tools	Alavi, 1984a	Field Study	Need special tools
	Burns and Dennis, 1985	Conceptual	Need special tools only for evolutionary prototyping
	Carey and Mason, 1983	Case Study	No special tools needed for expendable prototyping
	Dennis, Burns, Gallupe, 1987	Case Study	Need special tools only for evolutionary prototyping
	Gilhooley, 1987	Conceptual	Need special tools
	Klingler, 1988	Conceptual	Need special tools
	Mahmood, 1987	Survey	Need special tools
	Naumann and Jenkins, 1982	Conceptual	Need special tools
	Saarinen, 1990	Survey	Need special tools
	Schultz and Eierman, 1987	Survey	No special tools needed

Only a few empirical studies have investigated the influence of prototyping on system success. In two related studies, a field study and a lab study, Alavi (1984a, 1984b) examined the effect of design approach - prototyping versus life cycle - on the success of a system as measured by user satisfaction, accuracy, ease of use, and usefulness of output. The only conclusive result indicated users were more satisfied with the system produced with prototyping. Another lab study of seven development teams, designed to compare the SDLC and prototyping, found that the prototyped products were 40 percent smaller and required 45 percent less effort to develop (Boehm, Gray, and Seewaldt, 1984). Alter and Ginzberg (1978), in a field study of 56 projects, found that prototyping could be used to reduce risk, and thus increase system success.

Anecdotal evidence via case studies also indicates the positive influence of prototyping on system success. Kraushaar and Shirland (1985) evaluated successful implementation under the conditions of prototyping versus life cycle via case studies. Their results indicated that prototyping can provide on-time and within-budget systems. Berrisford and Wetherbe (1979) provide several successful case studies of companies that have used prototyping. The studies illustrate some positive effects, including shorter development time and greater user satisfaction. Cost savings and greater user satisfaction have also been demonstrated in other case studies (Asner and King, 1981; Groner, Hopwood, Palley, and Sibley, 1979; Johnson, 1983).

Several surveys have indicated the successful implementation of systems developed using a prototyping approach. Mahmood's survey (1987), designed to test the impact of development method on system success, found that prototyping increased the use of the system by users, and user satisfaction was higher when using the prototyping approach (although not statistically significant). Other surveys by Langle, Leitheiser, and Naumann (1984), Guimaraes and Saraph (1991), and Necco, Gordon, and Tsai (1987), indicated that prototyping resulted in higher user satisfaction and lower development time, compared to conventionally developed systems. Seventy-seven percent of the respondents to Schultz and Eierman's (1987) survey indicated savings from one to 60 percent of the design time over the traditional SDLC. Twenty-five percent indicated a savings of 41 to 60 percent.

As stated several times throughout this study, prototyping is not a panacea for all development projects. Thus, if not used properly, prototyping can result in project failure. Several case studies provide evidence to the previous statement.

In a study of nine projects using prototyping (two used an evolutionary prototyping strategy, seven used an expendable prototyping strategy), all the projects were considered failures because they well exceeded budget and schedule constraints (Gronbaek, 1989). One of the systems, which used an expendable prototyping strategy, was abandoned before it was completed. Specific reasons for the failures were not provided. In a case study presented by Berrisford and Wetherbe (1979), users refused to allow the development group to dismantle an expendable prototype because the users did not understand *a priori* that the prototype was not for production use. As a result, the users were not initially satisfied with the system.

Another system failure using prototyping was reported by Connell and Brice (1985). An evolutionary prototyping strategy was selected as the development approach for a project which was to replace an existing system. The new system was eventually scrapped after the development effort became far behind schedule and over-budget. The older system was re-implemented. Factors contributing to the failure include: (1) the developers inexperience with both prototyping and the prototyping tools; and (2) many users were involved in the development. In a similar case study involving evolutionary prototyping (Carey, 1990), a fourth generation language (with evolutionary prototyping) was used to develop the State of New Jersey Division of Motor Vehicles information system. When delivered, the response times were so slow that the backlogs resulted in thousands of motorists driving with invalid registrations or licenses. For this particular project, performance requirements were stringent, it was a critical system, and developers did not have experience with the development tool. Finally, the results of a survey conducted by Mahmood (1987) found that the number of jobs completed on schedule when the SDLC was used was significantly higher than when a prototyping approach was used.

2.4 Development of the Conceptual Model

The conceptual model developed from the factors previously discussed is shown in Figure 1. The model describes the relationships to be investigated in this study: the relationship between the various project, user, developer, and organization characteristics and the selection of a prototyping strategy; and the relationship between the prototyping strategy and system success. Two research questions are derived from Figure 1:

1. What factors influence the selection of a prototyping strategy, and are organizations using the contingencies suggested in the literature?
2. Does the proper use of a prototyping strategy, as suggested by the contingencies, influence system success?

2.5 Contingency Models

This section provides an overview of contingency theory. Because this study is exploratory, contingency theory is used to frame the study. As evidence of the importance of contingency theory in IS research, a

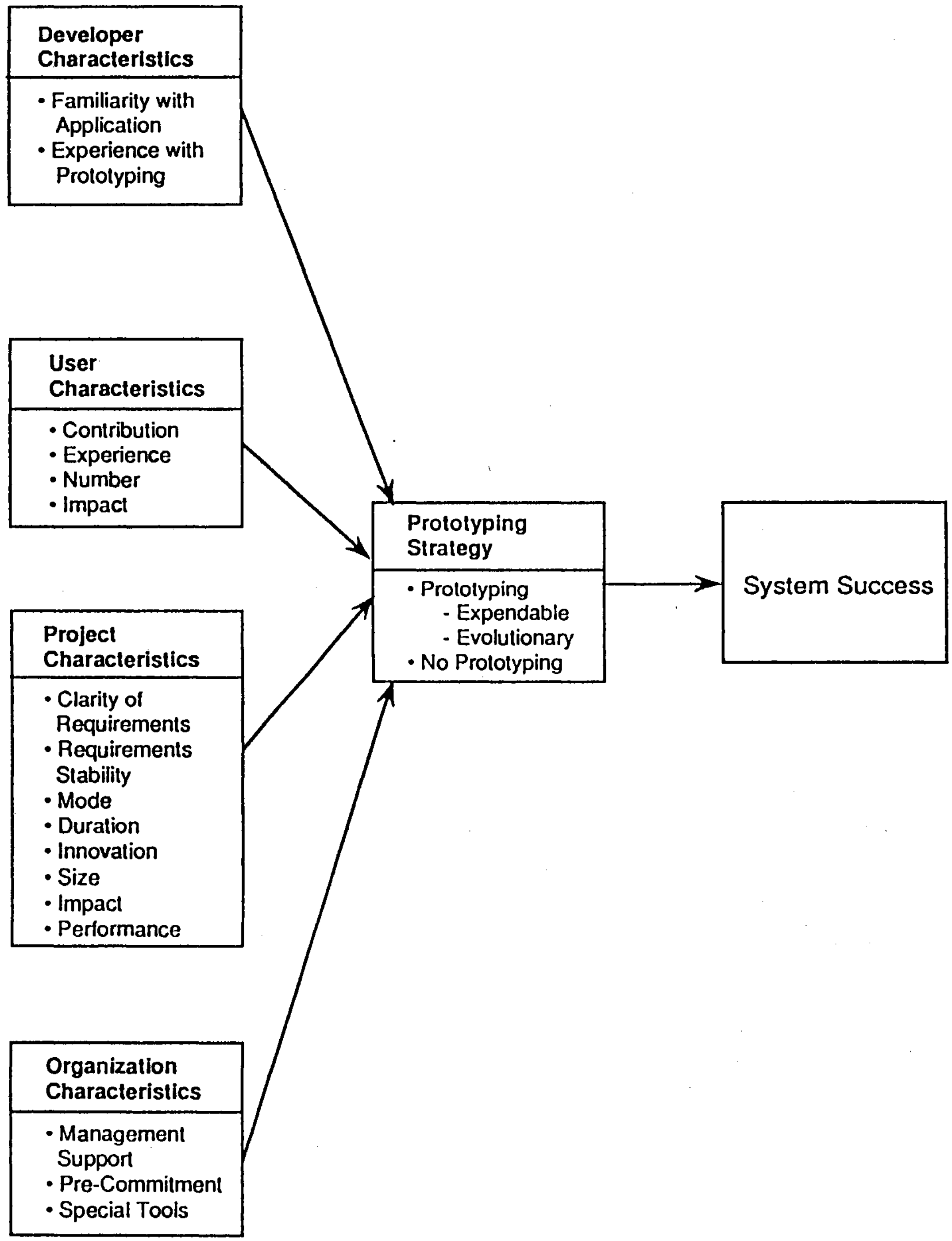

Figure 1. Conceptual Model of Prototyping Strategy Selection

discussion of the use of contingency models in software development research is also provided.

2.5.1 An Overview of Contingency Theory

Contingency theory can be viewed as a framework for organizing knowledge in a given area; alone, contingency theory has no content. Contingency theory merely acknowledges that certain variables may affect the outcome of a particular process. Contingency theory recognizes a functional relationship (not necessarily cause and effect) between independent variables and dependent variables. The goal of developing the contingency relationship is to achieve the most effective solution possible. For working purposes, the contingency theory is translated into if-then terms. The if's represent the independent variables and the then's represent the dependent variables in the functional relationship. In other words, if certain conditions exist, then certain concepts and techniques are more effective than others for goal attainment (Luthans, 1976). For example, one contingency developed in this study may be: "if the system mode is on-line, then use prototyping."

The assumptions underlying contingency theory are (Weill and Olson, 1989):

1. Fit: contingency theory assumes that the better the "fit" among contingency variables, the better the performance of the organization.

2. Performance: performance may or may not be measured in the study; when it is, it is generally narrowly defined by quantitative measures.
3. Rational actors: the theory assumes that organizational actors perform in ways that are always in concert with the superordinate goal of organizational effectiveness.
4. Equilibrium: an organization with "fit" is at equilibrium, and performance is a result of that equilibrium. Thus, there is no time lag between the independent variables and their impact on organization performance (functional relationship).
5. Deterministic model: the methodologies employed do not generally allow conclusions about causality; although causal inference can be made.

Contingency theory is widely used in IS research. The contingency approach to IS suggests that a number of variables influence the performance of information systems; the better the "fit" between these variables and the design and use of the MIS, the better the IS performance. A study of research published between 1982 and 1988 in MIS Quarterly and the Journal of Management Information Systems found that over 70 percent of the reported empirical studies used a contingency model (Weill and Olson, 1989).

2.5.2 Information Systems Development Contingency Models

The application of contingency theory to information system development is not new. Various models have evaluated different aspects of software development, and the factors affecting it. A brief summary of a few of the models follows.

Alter and Ginzberg (1978) used a contingency model to select a development strategy based upon its ability to reduce project risk. Contingency models for selecting a development strategy based upon information needs (Schonberger, 1980), type of organization (Gibson, Singer, Schnidman, and Davenport, 1984), project uncertainty (Burns and Dennis, 1985; El Louadi, Pollalis, and Teng, 1991), project complexity (Burns and Dennis, 1985; El Louadi, Pollalis, and Teng, 1991), system type (Cerveny, Garrity, and Sanders, 1986; El Louadi, Pollalis, and Teng, 1991), project duration (Dos Santos, 1986, 1988), project size (Lynch, 1987), project structure (Lynch, 1987), user requirements (Dos Santos, 1986, 1988), system impact (Dos Santos, 1986, 1988), and analyst exposure to technology (Lynch, 1987), have also appeared.

Naumann, Davis, and McKeen (1980) developed a contingency model for selecting an information requirements analysis (IRA) approach (in their study, prototyping is considered an IRA method). The variables in this model included project size, degree of structure, user task comprehension, and developer task proficiency. Davis (1982) proposed a similar model which used the characteristics of the utilizing system, the information system or application, the users, and the analysts, as the contingencies. In the previous two models, IRA method was the dependent variable. However, Sethi and Teng (1988) used the IRA method as the independent variable in choosing the finishing strategy for systems development.

Many of the aforementioned models are purely conceptual and have not been validated. In one of the few empirical studies, Saarinen (1990) conducted a survey in an attempt to validate some of the more common contingencies. Surprisingly, results did not substantiate the majority of the contingencies tested.

2.6 Chapter II Summary

This second chapter has presented a review of the literature necessary for the development of this study. As discussed in Chapter I, one purpose of this study is to determine which factors influence the selection of a prototyping strategy. As such, the majority of this chapter was devoted to identifying those factors from the prototyping literature. Seventeen factors, in four categories, were identified and discussed.

The conceptual model, in addition to identifying the factors affecting prototyping strategy, also demonstrates the effect of prototype selection on system success. Thus, a brief review of system success was provided. Additionally, empirical studies which have investigated the effect of prototyping on system success, and system failure, were provided.

Next, an overview of contingency theory was presented. The definition and importance of contingency theory, and its application to information systems development situations has been discussed. A review of some previously developed contingency models was provided.

CHAPTER III

RESEARCH METHODOLOGY

3.1 Overview

The emphasis of this chapter is on the research methodology used in this study. First, the propositions and hypotheses are derived and presented. Second, the sample, data collection method, and task are discussed. Third, operationalizations of the variables are provided, along with the creation of the questionnaire used to collect the data. Lastly, a preliminary discussion of possible data analysis approaches is provided.

3.2 Propositions and Hypotheses

This study is exploratory in the sense that little previous research has been conducted in the area of prototyping strategy selection, and no known theory of prototyping strategy selection exists. From the available literature, a conceptual model of prototyping strategy selection has been developed to serve as a framework to guide this study (see Figure 1). From the conceptual model, five propositions are gleaned: propositions examining the relationship between each of the four sets of characteristics and prototyping strategy selection; and a proposition investigating the relationship between prototyping selection and system success. The propositions are:

Proposition 1: The selection of a prototyping strategy is related to project characteristics.

Proposition 2: The selection of a prototyping strategy is related to user characteristics.

Proposition 3: The selection of a prototyping strategy is related to developer characteristics.

Proposition 4: The selection of a prototyping strategy is related to organization characteristics.

Proposition 5: The success of a system is related to the selection of a prototyping strategy.

Propositions are truth statements about a model and, by nature, are not testable (Dubin, 1969). However, hypotheses derived from the propositions are testable. Hypotheses are statements of predictions of what will be true in the real world if the evidence from the real world is supportive (Dubin, 1969). In the following sections, the specific hypotheses used to test the five propositions are provided. Tables 5 through 9 summarize the propositions and related hypotheses.

For this study, the lack of theoretical or empirical evidence requires the hypotheses to be non-directional. Therefore, the hypotheses can suggest a relationship between the variables, but cannot suggest causality. Each hypothesis is stated according to the contingency view of "if-then." Although an "else" is not provided for each hypothesis, it is implied.

As a reminder, the generic term "prototyping" indicates the selection of either an expendable or evolutionary prototyping strategy. For convenience, the corresponding section from Chapter 2 is noted for each hypothesis.

3.2.1 Project Characteristics: Hypotheses

A prototype, expendable or evolutionary, can help clarify the needs of the users and requirements of the system when requirements are unclear, or users are vague or ambiguous (Alavi, 1984a; Asner and King, 1981; Berrisford and Wetherbe, 1979; Burns and Dennis, 1985; Carey and Currey, 1989; Connell and Brice, 1985; Davis, 1982; Dos Santos, 1986, 1988; Gavurin, 1991; Gremillion and Pyburn, 1983; Janson and Smith, 1985; Kauber, 1985; Kraushaar and Shirland, 1985; Mahmood, 1987; Naumann and Jenkins, 1982; Naumann, Davis, and McKeen, 1980; Slusky, 1987; Smith, 1987; Yaverbaum, 1989). Prototyping may increase development time when specifications are unambiguous by involving users more than necessary (Gavurin, 1991). Although there are those who do not agree with this contingency (Carey, 1990; Krzanik, 1986; Lynch, 1987), they are a very small minority. Therefore:

H1: If requirements are unclear, then prototyping will be used (see Section 2.2.1.1).

A system whose requirements are expected to change during development requires a development approach that can accommodate the changes (Klingler, 1986; Kraushaar and Shirland, 1985; Li, 1990; Naumann and Jenkins, 1982; Smith, 1987). A prototype, expendable or evolutionary, can accommodate change. Therefore:

> H2: If the system requirements are dynamic, then prototyping will be used (see Section 2.2.1.2).

On-line systems require more interaction between user and developer because of the associated user interfaces. On the other hand, batch systems require very little interaction because of the lack of a user interface. Prototypes, expendable or evolutionary, can better obtain specifications from users when user interfaces are needed (Burns and Dennis, 1985; Carey, 1990; Carey and Currey, 1989; Gavurin, 1991; Graham, 1989; Klingler, 1986, 1988; Mahmood, 1987; Mason and Carey, 1983; Necco, Gordon, and Tsai, 1987; Smith 1987). Therefore:

> H3: If the system mode is on-line, then prototyping will be used (see Section 2.2.1.3).

The longer the duration of a project, the greater the changes are likely to be, and the greater the risk, that the project will become unmanageable. This argument would suggest that, due to problems associated with managing prototyping, prototyping should not be used for projects of long duration (Dos Santos, 1986, 1988; Li, 1990; Smith, 1987). Therefore:

> H4: If the duration is long, then prototyping will not be used (see Section 2.2.1.4).

New system development (high innovation) requires a higher user/developer interaction than modifications to existing systems (low innovation). Thus, it would seem appropriate to use prototyping, which facilitates the interaction between user and developer (Carey and Currey, 1989; Dos Santos, 1988; Johnson, 1983). Therefore:

> H5: If innovation is high, then prototyping will be used (see Section 2.2.1.5).

Project size is an issue of debate in prototyping selection. Of the more than 20 studies cited, the number of those suggesting the use of prototyping for large systems (Brittan, 1990; Dearnley and Mayhew, 1983; Gavurin, 1991; Groner, Hopwood, Palley and Sibley, 1979; Guimaraes, 1981; Gupta, 1988; Johnson, 1983; Kraushaar and Shirland, 1985; Pliskin and Shoval, 1987; Saarinen, 1990; Schultz and Eierman, 1987; Shoval and Pliskin, 1988), and those suggesting the use of prototyping for small systems (Burns and Dennis, 1985; Carey, 1990; Dennis, Burns, and Gallupe, 1987; Klingler, 1986, 1988; Kraushaar and Shirland, 1985; Lynch, 1987; Mahmood, 1987; Pliskin and Shoval, 1987; Saarinen, 1990; Shoval and Pliskin, 1988; Yaverbaum, 1989), is almost equal. However, of the aforementioned studies, only seven are empirical. Of the seven, four suggest using prototyping for large systems (Brittan, 1990; Groner, Hopwood, Palley, and Sibley, 1979; Johnson, 1983; Schultz and Eierman, 1987), one suggests using prototyping for small systems (Dennis, Burns, and Gallupe, 1987), and two suggest that size is not a consideration (Kraushaar and Shirland, 1985; Saarinen, 1990). Therefore, based on the majority of the empirical evidence:

> H6: If project size is large, then prototyping will be used (see Section 2.2.1.6).

For critical systems - systems that operate, manage, and control the daily business activities - it is extremely important that the system meet specifications. Prototyping facilitates requirements (specifications) determination, and should, therefore, be used for critical systems (Boar, 1986; Carey, 1990; Dos Santos, 1988; Gremillion and Pyburn, 1983). Therefore:

> H7: If the system is a critical system, then prototyping will be used (see Section 2.2.1.7).

Evolutionary prototyping has been criticized because of the poor performance of the final system (Connell and Brice, 1984; Swift, 1989). The special tools needed to develop an evolutionary prototype are usually not adequate to support a system with a heavy volume of transactions, or a high performance requirement. Therefore:

H8: If a system has stringent performance requirements, then an evolutionary prototyping strategy will not be used (see Section 2.2.1.8).

TABLE 5

PROPOSITION AND HYPOTHESES FOR PROJECT CHARACTERISTICS

Proposition 1: The selection of a prototyping strategy is related to project characteristics.

H1: If requirements are unclear, then prototyping will be used.

H2: If the system requirements are dynamic, then prototyping will be used.

H3: If the system mode is on-line, then prototyping will be used.

H4: If the duration is long, then prototyping will not be used.

H5: If innovation is high, then prototyping will be used.

H6: If project size is large, then prototyping will be used.

H7: If the system is a critical system, then prototyping will be used.

H8: If a system has stringent performance requirements, then an evolutionary prototyping strategy will not be used.

3.2.2 User Characteristics

Prototyping, compared to the SDLC, requires more communication and interaction between the user and developer. If the users do not dedicate time to the project, then the use of a prototype is not feasible (Gibson and Rademacher, 1987; Meyer and Kovacs, 1983; Teng and Sethi, 1990). Therefore:

> H9: If the users do not have time to dedicate to the project, then prototyping will not be used (see Section 2.2.2.1).

Users not familiar with the concept of prototyping often have unrealistic expectations regarding the speed of delivery and functionality of the system because a prototype may be seen as the operational system (Alavi, 1984a; Sroka and Rader, 1986). As a result, users are often dissatisfied with the system when it is not delivered immediately after viewing a prototype. Therefore:

> H10: If the users are inexperienced with prototyping, then prototyping will not be used (see Section 2.2.2.2).

The previous hypothesis assumes the user has experience with automation, but no experience with prototyping. However, if the user has no experience with automation, prototyping can be used to help the user develop a better idea of their system requirements (Ahituv, Hadass, and

Neumann, 1984; Davis, 1982; Dearnley and Mayhew, 1983; Gavurin, 1991; Tate and Verner, 1990). Therefore:

> H11: If the users have no automation experience, then prototyping will be used (see Section 2.2.2.2).

Involving a large number of users in a project employing prototyping can increase the project development time (Burns and Dennis, 1985; Connell and Brice, 1985; Gavurin, 1991; Tillman, 1989). The idea behind prototyping is to accommodate the needs and wishes of the user. If several users are requesting alterations to the prototype, it becomes difficult to manage the changes. Therefore:

> H12: If there are a large number of users, then prototyping will not be used (see Section 2.2.2.3).

If the user impact is high (i.e., the project will considerably change the way users perform their jobs), there should be a high degree of interaction between the user and the developer. A high interaction will keep the user informed and reduce the likelihood of system rejection. Since a prototyping approach requires a large amount of interaction between the developer and the user, it would appear to be appropriate in cases of high user impact (Asner and King, 1981; Dos Santos, 1988). Therefore:

> H13: If the user impact is high, then prototyping will be used (see Section 2.2.2.4).

TABLE 6

PROPOSITION AND HYPOTHESES FOR USER CHARACTERISTICS

Proposition 2:	The selection of a prototyping strategy is related to user characteristics.
H9:	If the users do not have time to dedicate to the project, then prototyping will not be used.
H10:	If the users are inexperienced with prototyping, then prototyping will not be used.
H11:	If the users have no automation experience, then prototyping will be used.
H12:	If there are a large number of users, then prototyping will not be used.
H13:	If the user impact is high, then prototyping will be used.

3.2.3 Developer Characteristics

When the application domain is familiar to the developers, interaction with the users during development may not be necessary (Kraushaar and Shirland, 1985; Naumann, Davis, and McKeen, 1980). Unnecessary user involvement can increase development time (Kraushaar and Shirland, 1985). Prototyping, expendable or evolutionary, requires interaction with the user. Also, for developers lacking experience with the application, prototyping allows for more interaction with the user which increases the degree of certainty that the developer will accurately and completely elicit and document the user's requirements (Alter and Ginzberg, 1978; Naumann, Davis, and McKeen, 1980). Therefore:

> H14: If developers have experience with similar applications, then prototyping will not be used (see Section 2.2.3.1).

Prototyping facilitates change during the development process, the SDLC does not. Due to the frequent changes in user requirements, designers accustomed to the SDLC may find prototyping frustrating unless they are prepared to expect changes and view them positively (Alavi, 1984a; Sroka and Rader, 1986). Also, prototyping requires a greater level of interpersonal communication skills from the developers (Alavi, 1984b). Therefore:

> H15: If developers are not experienced with prototyping, then prototyping will not be used (see Section 2.2.3.2).

3.2.4 Organization Characteristics

Prototyping requires a new way of approaching the entire system development process. Planning and controlling prototyped systems are more difficult to manage because of their evolving nature, the number of revisions, and the uncertainty in user requirements. Failure to gain management support and establish explicit procedures can lead to frustration from both users and developers, and can result in system failure (Alavi, 1984a; Klingler, 1988; Voltmer, 1989). Therefore:

H16: If a project does not have management support, then prototyping will not be used (see Section 2.2.4.1).

TABLE 7

PROPOSITION AND HYPOTHESES FOR DEVELOPER CHARACTERISTICS

Proposition 3: The selection of a prototyping strategy is related to developer characteristics.

H14: If developers have experience with similar applications, then prototyping will not be used.

H15: If developers are not experienced with prototyping, then prototyping will not be used.

Prototyping provides the opportunity to test, modify, and visualize a real-life system without the resources needed for the full project. Thus, prototyping can be used in situations where there is a need for experimentation and learning before commitment of resources for a full project (Alavi, 1984a, 1984b; Asner and King, 1981; Mahmood, 1987; Smith, 1987). Therefore:

H17: If a need for experimentation and learning before full commitment exists, then prototyping will be used (see Section 2.2.4.2).

Special tools, such as DBMS, 4GLs, and application generators, are advancing the use of prototyping. For evolutionary prototyping, special tools are needed to facilitate fast response to user requests (Alavi, 1984a; Burns and Dennis, 1985). However, expendable prototypes can be built without the use of any special tools, such as word processors (Carey and Mason, 1983). Therefore:

H18: If prototyping support tools are not available, then an evolutionary prototyping strategy will not be used (see Section 2.2.4.3).

TABLE 8

PROPOSITION AND HYPOTHESES FOR ORGANIZATION CHARACTERISTICS

Proposition 4:	The selection of a prototyping strategy is related to organization characteristics.
H16:	If a project does not have management support, then prototyping will not be used.
H17:	If a need for experimentation and learning before full commitment exists, then prototyping will be used.
H18:	If prototyping support tools are not available, then an evolutionary prototyping strategy will not be used.

3.2.5 System Success: Hypothesis

The appropriate use of prototyping, expendable or evolutionary, has a positive influence on the success of a system. This statement is consistent with previous findings (Alavi, 1984a, 1984b; Alter and Ginzberg, 1978; Asner and King, 1981; Berrisford and Wetherbe, 1979; Boehm, Gray, and Seewaldt, 1984; Groner, Hopwood, Palley, and Sibley, 1979; Guimaraes, 1981; Guimaraes and Saraph, 1991; Johnson, 1983; Kraushaar and Shirland, 1985; Langle, Leitheiser, and Naumann, 1984; Mahmood, 1987; Necco, Gordon, and Tsai, 1987; Saarinen, 1990). Therefore:

> H19: A properly selected prototyping strategy is positively related to system success (see Section 2.3).

TABLE 9

PROPOSITION AND HYPOTHESIS FOR SYSTEM SUCCESS

Proposition 5:	The success of a system is related to the selection of a prototyping strategy.
H19:	A properly selected prototyping strategy is positively related to system success.

3.3 Research Methodology

3.3.1 Sample

The sample for this study is 500 systematically selected firms from the Directory of Top Computer Executives from Applied Computer Research. This directory lists over 13,500 IS executives in organizations throughout the United States. To obtain a sample of 500, a systematic sampling technique is used. A random number is used to provide the starting position, then every seventeenth name is selected (i.e., 13,500 / 500 = 27). This type of sampling should provide a random sample of organizations, and is expected to provide a broad cross-section of firms by industry. Using a similar sampling strategy, Necco, Gordon, and Tsai (1987) obtained results from a wide variety of firms. A justification for the sample size is provided in a later section.

3.3.2 Data Collection Strategy

This study uses a mail survey of multiple organizations and their recently implemented information systems to investigate the hypotheses summarized previously in Tables 5 through 9. This research strategy follows those used in similar studies (e.g., Mahmood, 1987; Moore, 1979; Necco, Gordon, and Tsai, 1987; Olson and Ives, 1981; Tait and Vessey, 1988).

The mail survey technique will be conducted in accordance with accepted survey practices (Fowler, 1988; Rea and Parker, 1992). The questionnaire packet will be pre-tested with a small group of IS departments which have agreed to participate. The results of the pre-test will be used to modify the questionnaire, if necessary. The data obtained from the pre-test will not be used in the final analysis. The cover letter and pre-tested questionnaire are provided in Appendix A.

A preliminary notification letter will be sent to the IS manager at each of the selected organizations. The notification is used to inform the IS manager of the nature of the study, and asking for his/her cooperation. Approximately four days after the preliminary notification, the questionnaire packet will be sent. The questionnaire packet will consist of a cover letter, two copies of a developer questionnaire, two copies of a user questionnaire, and four postage-paid return envelopes (one for each questionnaire). Copies of the preliminary letter, cover letter, and questionnaires are contained in Appendix B.

The IS manager will be asked to select at least two systems (one for each set of questionnaires) that have been implemented during the last two years. The systems can be either "good" or "bad." This type of selection is used to increase the probability that successful and unsuccessful systems are equally, or at least partially, represented (Vroom and Jago, 1978). The IS managers are then asked to provide copies of the questionnaire to both the project leader or a key developer, and the primary user of the system, for each of the two systems. The administration of separate questionnaires should avoid the problem described by Cook and Campbell (1979) of having the same individual provide all sources of information (i.e., mono-methods bias).

Since 95 percent of all returns should be received within the first three to four weeks (Alreck and Settle, 1985), a reminder/thank-you card will be sent during the third week after the initial mailing. The reminder/thank-you card is used to boost the response rate. Again, during the third week after the reminder, a last mailing containing the questionnaire packet will be sent to non-respondents. The entire process of data collection will take approximately two months, which is normal for a mail survey (Fowler, 1988).

3.3.2.1 Appropriateness of a Mail Survey

A survey-based study is the most appropriate form of data collection method for this study for several reasons. First, according to Galliers (1985), survey research should be used to: (1) study IS failures or implementation efforts; or (2) study the impact of information technology and IS on organizations; or (3) study the role and effects of information technology and IS on society. The study of IS development approaches is contained within Galliers' first category - clearly the study described herein fits into the first category. Surveys provide the opportunity to study a greater number of variables necessary in IS development approaches. Lab experiments and field experiments do not provide the breadth of variables required (Galliers, 1985).

Second, development duration makes it difficult to study IS development with methods other than survey. Computer systems can have a development period of several years, making direct observations, or realistic lab experiments, infeasible (Edstrom, 1977; Schach, 1990). Third, companies are unwilling to produce systems for the sake of experiments, thus eliminating the field experiment option (Edstrom, 1977).

Fourth, surveys are used extensively in IS research. A survey of MIS researchers found that 23 percent of the in-progress projects were utilizing survey research (Teng and Galletta, 1990). Mean endorsement rankings from the aforementioned survey indicate that survey research is the most highly rated method. Also, Orlikowski and Baroudi (1991), in a study of the leading MIS publication outlets (Communications of the ACM, MIS Quarterly, Management Science, and ICIS Proceedings) from 1983 to 1988, found that 49.1 percent of the articles used survey research, followed by 27.1 percent for lab studies, and 13.5 percent for case studies. Lastly, in a survey of published contingency studies from 1982 to 1988 in MIS Quarterly and Journal of

Management Information Systems, 67 of the 74 contingency studies used a survey methodology (Weill and Olson, 1989).

In addition to those previously mentioned specifically for IS research, mail surveys offer the following benefits. First, there is no better method of research than the sample survey process for determining, with a known level of accuracy, information about large populations (at a realistic cost) (Fowler, 1988; Kerlinger, 1973; Rea and Parker, 1992). Second, the questionnaire can be completed at the respondent's convenience and at their own pace (Rea and Parker, 1992). Third, because their is no personal contact with the interviewer, the respondents have a feeling of anonymity (Rea and Parker, 1992). Fourth, mail surveys reduce interviewer bias (Rea and Parker, 1992).

There are, however, criticisms of the survey approach to data collection. One of the most frequently mentioned problems with a mail survey is the low response rate (Kerlinger, 1973; Rea and Parker, 1992). It is not uncommon, in IS research, to see response rates as low as 14 percent (Langle, Leitheiser, and Naumann, 1984), or as high as 48 percent (Carey and McLeod, 1988). In this study, two questionnaires are required for each project, therefore, it is suspected that the response rate will suffer. However, Mahmood (1987) used the strategy of sending a questionnaire to a user and a developer and received an 18 percent response rate. By following a process of multiple follow-ups, which will boost response, the response rate should be adequate for this study.

Another problem associated with survey research is non-respondent bias (Kerlinger, 1973; Rea and Parker, 1992). Although it is hard to prevent non-respondent bias, we have two methods of testing for its existence (Oppenheim, 1966). The first is to compare respondents against non-respondents according to demographics. The second way is to compare early respondents with late respondents (in terms of their answers to questions). It has been found that late respondents are roughly equivalent to non-respondents (Oppenheim, 1966).

3.3.3 Independent and Dependent Variable(s)

One relationship depicted by the conceptual model (see Figure 1) is the selection of a prototyping strategy as a function of the project characteristics, user characteristics, developer characteristics, and organization characteristics. Therefore, in this case, the prototyping strategy is the dependent variable and the four categories of characteristics (17 factors) are the independent variables.

The second relationship illustrated by the conceptual model indicates that system success is a function of the prototyping strategy. Therefore, system success becomes the dependent variable and the prototyping strategy is the independent variable.

The switching of the prototyping strategy from dependent variable to independent variable, based upon its function in the model, is consistent with work by Vroom and Yetton (1973, p. 198). In Vroom and Yetton's model, situational variables and personal attributes are independent variables and leader behavior is the dependent variable. Additionally, leader behavior is the independent variable when organizational effectiveness is the dependent variable.

It should be noted that the terms "independent" and "dependent" variables are used loosely. For a model, such as the one used in this study, which does not show causality, the classification of variables as independent and dependent is not entirely accurate. However, in contingency theory, the terms are used to clarify the relationships. As mentioned in Chapter 2, in a contingency statement the if's represent the independent variables and the then's represent the dependent variables in the functional relationship. That is, if certain conditions exist, then certain concepts and techniques are more effective than others for goal attainment (Luthans, 1976).

3.3.4 Measurement of the Model Variables

Each of the independent and dependent variables will need to be operationalized to be useful. Operationalization provides a way to measure the variables. In this study, the measures will be obtained from respondent's answers to the questionnaire. When possible, previously used measures of the variables will be utilized.

The following sections will develop the questionnaire used in this study as a result of the operationalizing of the variables. The complete questionnaire is provided in Appendix B.

3.3.4.1 Project Characteristics

3.3.4.1.1 Clarity of Requirements. One question, adopted from Gremillion and Pyburn (1983), is used to determine the clarity of the requirements prior to the development effort: "Were the inputs and outputs required of the system specified in advance?" Response is based on a seven-point scale with the following anchors: (1) specified completely; (7) not specified.

3.3.4.1.2 Requirements Stability. To determine the stability of the requirements after initiation of development, one question is used: "Did the system requirements change after development started?" The response is measured on a seven-point scale ranging from: (1) no changes; to (7) many changes.

3.3.4.1.3 System Mode. System mode is easily determined by one direct question: "Is the system on-line or batch?" Respondents are provided with three choices: (1) on-line, (2) batch, and (3) elements of both.

3.3.4.1.4 Duration. To measure system development time, an open-ended question, similar to a question used by McFarlan (1981) and Srinivasan and Kaiser (1987), is used: "How much time, in months, was needed to develop the system (time to develop includes analysis, design, construction, testing, and implementation)?".

3.3.4.1.5 Innovation. One question can be used to measure the innovation level of the system: "Did the system represent a new development effort, or a modification (i.e., redesign, enhancement, etc.) to an existing system?" Responses can be measured on the following scale: (1) new development; to (7) minor modification to existing system.

3.3.4.1.6 Size. Size is a relative term. That is, a $50,000 project is small for a billion dollar company, but large to a small business. Additionally, size has several meanings. In this study, size indicates developer man-hours and/or development cost. Therefore, a four-item instrument, based partly on work by Jenkins, Naumann, and Wetherbe (1984), McFarlan, McKenney, and Pyburn (1983), and McFarlan (1981), is used. The four questions are: "What was the original budget for the system?", "What is the annual budget of the Information Systems department?", "How many total development man-hours did the system require (development includes analysis, design, construction, testing, and implementation)?", and "How does this system rank, in cost, compared to other systems developed during the last two years?" The first three questions are open-ended; the last question is measured on a seven-point scale with the following anchors: (1) upper 10%; (7) lower 10%. The first two questions are used to calculate a percentage-of-overall-budget used by the project, which can be used as an indicator of size.

3.3.4.1.7 Impact. Impact is measured by one question: "Does the system operate, manage, and control the daily business activities of the organization?". The response is a simple "yes" or "no."

3.3.4.1.8 Performance. Determining the performance standards of the system requires only one question: "How would you describe the performance requirements of the system? (e.g., response time, throughput)". Response is based on a seven-point scale with the following anchors: (1) strict requirements; (7) low requirements.

3.3.4.2 User Characteristics

3.3.4.2.1 Contribution. An adaptation of a question used by Guimaraes, Igbaria, and Lu (1992), Fuerst and Cheney (1982), and Sanders and Courtney (1985), is used to measure the participation of the user in the development process. The participant is asked to respond to the following question: "In terms of your participation with the developers of this system, you have ...." (1) participated to a great extent; to (7) did not participate.

3.3.4.2.2 Experience. Experience is measured by two items. The first is used to measure experience with automation, the second to measure experience with prototyping. Both questions are based on McFarlan (1981). To indicate automation experience, the question "Prior to this system, how much experience did you have in using computerized systems?" is directed toward the user. The responses range from: (1) no experience to (7) extensive experience.

For prototyping experience, users are asked to respond to an answer from the following range - (1) no knowledge, to (7) extensive knowledge - to the question: "Prior to this system, how knowledgeable were you with the concept of prototyping?".

3.3.4.2.3 Number of Users. An open-ended question is used to indicate the number of users involved in the development effort: "How many users interacted with the development team during the development effort?".

3.3.4.2.4 User Impact. User impact is measured by two items based on McFarlan (1981). Each item requires the respondents to indicate the degree of change on a seven-point scale. The two items are, first "Did the system change the way you performed your job?", and second, "Did user departments have to reorganize to meet the requirements of the system?". The first item is anchored with (1) no changes and (7) many changes. The second item is anchored with (1) no reorganization and (7) major reorganization. The responses to the questions are averaged to provide a degree of user impact.

3.3.4.3 Developer Characteristics

3.3.4.3.1 Familiarity with Application Domain. Based on McFarlan (1981), a developer's familiarity with the application domain can be measured with one item. Participants are asked to respond to the following question: "Prior to this system, how experienced was the development team in the system's application area?", restricted by the following range of responses: (1) no experience, to (7) extensive experience.

3.3.4.3.2 Experience with Prototyping. The measurement of experience requires one item. Adopted from McFarlan (1981), this question will measure the developer's experience with the concept of prototyping. "How experienced are you with the use of prototypes?", requires a response within the range of: (1) no experience, to (7) extensive experience.

3.3.4.4 Organization Characteristics

3.3.4.4.1 Management Support. Management support is measured by a two-item scale based on Guimaraes, Igbaria, and Lu (1992), and Sanders and Courtney (1985). Each item requires the respondent to indicate their agreement or disagreement on a seven-point scale ranging from (1) strongly disagree to (7) strongly agree. The two statements are, first: "Prior to system development, top management felt that the time and resources needed for the development of this system was a wise investment," and the second, "In this organization, top management is strongly in favor of the concept of prototyping." Responses to the two questions are averaged to produce a total management score.

3.3.4.4.2 Pre-Commitment. One question can be used to determine the need for pre-commitment: "Was there a need for experimentation and learning before commitment of resources for a full system?" Response is a simple "yes" or "no."

3.3.4.4.3 Special Tools. Based on a question by Langle, Leitheiser, and Naumann (1984), the following question can be used to determine the use of special tools: "Which of the following tools were used for this system? (check all that apply):". The choices are: no tools, text editors/word processors, database management system, fourth generation language, other (please specify).

3.3.4.5 Prototyping Strategy

To determine the prototyping strategy used for a particular project, the developer will be given the definition of prototyping used in this study (see Section

1.2.1), and then asked the following: "Which ONE of the following statements describes the prototyping strategy used for this system?". Respondents will be asked to select one of the following: (1) only mockups of reports and screens were produced; (2) prototype simulates some functions, but does not use real data; (3) prototype performs some actual system functions and uses real data; (4) the prototype evolved into the final system; and (5) no form of prototyping was used. A response of "1," "2," or "3" indicates an expendable prototyping strategy; a "4" indicates an evolutionary prototyping strategy; and a "5" indicates that prototyping was not used.

3.3.4.6 System Success

User satisfaction, defined as the extent to which users believe the information system is available to them and meets their information requirements (Ives, Olson, and Baroudi, 1983), is the preferred indicator of system success (Tait and Vessey, 1988). DeLone and McLean (1992) and Cerullo (1980) indicate user satisfaction as the single most important and widely used success factor.

There are several different instruments for measuring user information satisfaction (UIS) (e.g., Bailey and Pearson, 1983; Ives, Olson, and Baroudi, 1983; McKeen, 1983). Instead of creating another instrument to measure user satisfaction, it was decided to use an instrument already developed and tested. The Ives, Olson, and Baroudi

(1983) (IOB-UIS) instrument was chosen for this study because it has been identified as the best available measure of user information satisfaction (Doll and Torkzadeh, 1988).

The 13-item IOB-UIS instrument is designed to measure three aspects of user satisfaction: satisfaction with the product, satisfaction with the EDP staff and services, and user knowledge and involvement. As suggested by Baroudi and Orlikowski (1988), the instrument should be modified to reflect the requirements of the specific situation. For this study, we are only interested in the satisfaction with the product. Therefore, the original 13-item scale has been reduced to four questions. Each question has two scales consisting of seven values, and assigned scores from -3 to +3. Each question is scored by taking the average of the two scores from the scales. Also, a global user satisfaction, as recommended by Baroudi and Orlikowski (1988) and Doll and Torkzadeh (1988), was added to the instrument. The global question can assist in validating the instrument for this study. The instrument used to measure UIS for this study is presented in Table 10.

3.4 Analyses

The analyses serve two purposes. The first is to evaluate the extent to which the hypothesized contingencies exist in practice, and second, whether prototyping strategy decisions consistent with the recommendations of the contingencies affect system success.

TABLE 10

USER INFORMATION SATISFACTION MEASURE

Please indicate the degree of congruence between what you wanted or required and what is provided by the information system (please circle one response from each scale).

1	2	3	4	5	6	7
extremely useful	quite useful	slightly useful	neither useful or useless	slightly useless	quite useless	extremely useless

1	2	3	4	5	6	7
extremely relevant	quite relevant	slightly relevant	neither relevant or irrelevant	slightly irrelevant	quite irrelevant	extremely irrelevant

Please indicate the correctness of the output from the information system (please circle one response from each scale).

1	2	3	4	5	6	7
extremely inaccurate	quite inaccurate	slightly inaccurate	neither accurate or inaccurate	slightly accurate	quite accurate	extremely accurate

1	2	3	4	5	6	7
extremely low	quite low	slightly low	neither high or low	slightly high	quite high	extremely high

Please indicate the consistency and dependability of the information from the system (please circle one response from each scale).

1	2	3	4	5	6	7
extremely high	quite high	slightly high	neither high or low	slightly low	quite low	extremely low

1	2	3	4	5	6	7
extremely superior	quite superior	slightly superior	neither superior or inferior	slightly inferior	quite inferior	extremely inferior

Please indicate the precision of the output from the system (please circle one response from each scale).

1	2	3	4	5	6	7
extremely high	quite high	slightly high	neither high or low	slightly low	quite low	extremely low

1	2	3	4	5	6	7
extremely definite	quite definite	slightly definite	neither definite or uncertain	slightly uncertain	quite uncertain	extremely uncertain

How would you rate your satisfaction with this system?

1	2	3	4	5	6	7
extremely low	quite low	slightly low	neither high or low	slightly high	quite high	extremely high

3.4.1 Testing Hypotheses for Prototyping Strategy Selection

Hypotheses requiring a dichotomous response can be tested by examining the relationship between each contingency's response and the selection of a prototyping strategy. Consider the following example of the test for Hypothesis 3 as an illustration of this technique. If the response to the question: "Is the system on-line or batch?" is "on-line", then "prototyping" is hypothesized to be chosen. We can now test the contingency against the outcome.

Four outcomes are possible: (1) an evolutionary or expendable strategy is chosen when the answer to the question is "batch" - this would contradict the hypothesis; (2) an evolutionary or expendable strategy is chosen when the answer is "on-line" - this is in agreement with the hypothesis; (3) "no prototyping" is chosen when the answer is "batch" - this is in agreement with the hypothesis; and (4) "no prototyping" is chosen when the answer is "batch" - this would contradict the hypothesis. These outcomes are expressed in Table 11. Fictitious numbers have been provided for illustration.

The appropriate statistic, based upon a 2 x 2 contingency table, is the Chi-squared statistic. As shown in Table 11, the significance level is 0.019. Thus, evidence suggests that H3 should not be rejected. This same procedure can be used for Hypotheses 7, 17, and 18.

TABLE 11

RELATIONSHIP BETWEEN HYPOTHESIS 3 AGREEMENT AND DECISION OUTCOME

		Prototyping	
		No	Yes
MODE	Batch	7*	5
	On-Line	1	9*

p-value = 0.019
* = correct choice

Hypotheses that do not require a dichotomous response (i.e., more than two categories or continuous) can be tested using analysis of variance (ANOVA). If must be noted that, although the analysis of variance technique requires the delineation of independent and dependent variables, only a statement of relationship will be concluded; causality is not assumed. For example, Hypothesis 4 requires a continuous-variable response (i.e., "How much time, in months, was needed to develop the system (time to develop includes analysis, design, construction, testing, and implementation)?"). The hypothesis states that a project with a long duration should use a "no prototyping" strategy. A t-test, a special case of ANOVA that applies to two distributions, is used to test the difference in the mean duration times between those projects using prototyping and those that do not. Table 12 provides an example illustrating this procedure. As seen in Table 12, evidence suggests that the hypothesis should not be rejected.

TABLE 12

RELATIONSHIP BETWEEN HYPOTHESIS 4 AGREEMENT AND DECISION OUTCOME

Prototyping?	N	Mean	Prob>|T|
NO	24	9.286	0.0001
YES	71	29.375	

3.4.2 Testing Hypothesis 19: System Success

Hypothesis 19 states that the proper selection of a prototyping strategy is positively related to system success. At this point, "proper selection" is based upon Hypotheses 1 through 18. Therefore, to test Hypothesis 19, it is necessary to examine the relationship between each specific hypothesis (1 through 18) and the success of the system. An example should clarify the explanation. Sample data is presented in Table 13 with the discussion following.

TABLE 13

RELATIONSHIP BETWEEN PROPERLY SELECTED PROTOTYPING STRATEGY (BASED ON HYPOTHESIS 3) AND SYSTEM SUCCESS

Correct?	N	Mean	Prob>|T|
NO**	45	-3.072	0.0001
YES*	46	9.027	

* YES = choose prototyping when system mode is on-line, or choose no prototyping when system mode is batch.

** NO = choose prototyping when system mode is batch, or choose no prototyping when system mode is on-line.

Using the example for Hypothesis 3 from the previous section (see Table 11), we can determine if the system success is higher when decisions are made in accordance to Hypothesis 3, than those that are not. For this example, the decision made in accordance with Hypothesis 3 is: a "use prototyping" strategy is chosen when system mode is on-line, or a "no prototyping" strategy is chosen when system mode is batch. Decisions other than these are not in agreement with the hypothesis. A t-test can then be used to test for significant differences between the means of the success values (obtained from the questionnaire) for those in agreement with Hypothesis 3 and those not in agreement. Each of the individual contingencies (specified by hypotheses 1 through 18) could be tested in a similar manner.

3.4.3 Building a Contingency Model of Prototyping Selection

One product of this study is a proposed contingency model of prototyping selection. Hypotheses 1 through 18 provide an indication of the factors affecting the selection of a prototyping strategy, and Hypothesis 19 indicates the impact on system success of the prototyping strategies. In addition to the findings from the hypotheses, we need to know how these factors are related before proposing the contingency model. One way of viewing the relationship among the variables is to determine their logical grouping, as suggested by Figure 1. The second way is to view the factor groupings in relation to prototyping strategy selection, and system success.

3.4.4 Instrument Validation

The validity of a questionnaire is the extent to which the questionnaire actually measures what it is supposed to measure (Bailey and Pearson, 1983; Baroudi and Orlikowski, 1988). Three types of validity that are generally accepted are (Nunnally, 1978): (1) predictive validity; (2) content validity; and (3) construct validity. In this study, previously validated instruments were used when possible. Thus, instrument validity should not be an issue, although it will be checked.

3.4.4.1 Predictive Validity

Predictive validity indicates the degree to which a measure predicts a second future measure. A high degree of predictive validity implies that the instrument is consistent and agrees with other independent measures (Bailey and Pearson, 1983). Due to the nature of this study, predictive validity is not an issue. The contingency model explored herein is a functional model, not a causal model. Thus, the predictive validity, in this instance, does not need to be investigated.

3.4.4.2 Content Validity

Content validity implies that all aspects of the concept being measured are considered by the instrument (Bailey and Pearson, 1983). An instrument is content valid if it has drawn representative questions from a universal pool (Straub, 1989).

Content validity is difficult to verify. Cronbach (1971) suggests a review process whereby experts in the field evaluate the instrument. This is a subjective indication of how well the questions measure the concept.

Circumstantial evidence can be provided by using correlation analysis. Scales which purport to measure the same attribute should be positively correlated (Bailey and Pearson, 1983; Ives, Olson, and Baroudi, 1983).

3.4.4.3 Construct Validity

Construct validity indicates the degree to which an operationalization of a variable actually captures the concept it purports to measure (Baroudi and Orlikowski, 1988; Sethi and King, 1991). Two methods of construct validation have been suggested (Nunnally, 1978): (1) correlation analysis; and (2) factor analysis.

Correlation analysis can be used to measure the degree of association between each item of the instrument and the total score (Baroudi and Orlikowski, 1988; Nunnally, 1978; Sethi and King, 1991). Each item is considered construct valid to the extent it correlates positively with the total score. For example, each of the four-items from the instrument used in this study to measure UIS can be correlated with the total score of the four instruments. Two forms of construct validity, based upon correlation analysis, are convergent validity and discriminant validity. Convergent validity is the extent to which a measure is correlated or agrees with other measures of the same construct (Baroudi and Orlikowski, 1988; Sethi and King, 1991). Discriminant validity is indicated by low correlations between the measure of interest and other measures not measuring the same concept (Baroudi and Orlikowski, 1988; Sethi and King, 1991).

The other method of construct validation is factor analysis. Factor analysis is a technique of finding clusters of related variables (Nunnally, 1978). Each cluster is denoted by a group of variables whose members correlate more highly among themselves than they do with variables not included in the cluster. For construct validity, factor analysis indicates that a group of items measure the same things; but not necessarily the right things.

3.4.5 Instrument Reliability

After determining the validity of an instrument, it is necessary to assess the reliability of the instrument. Reliability is the extent to which the questionnaire is free from measurement error (Baroudi and Orlikowski, 1988). Several methods are available to determine the reliability of an instrument. The statistical measures of reliability pertinent to this study are Kuder-Richardson 20 (commonly called KR-20), and coefficient alpha (commonly called Cronbach's alpha) (Nunnally, 1978). KR-20 is appropriate when there are only two responses for each item; Cronbach's alpha is used for items which are not scored dichotomously. Each of these methods is used to measure the internal consistency of an instrument which is administered one time. The amount of error in a measurement is determined by applying one of the methods to the inter-item scores and the overall measure (Nunnally, 1978).

3.4.6 Justification for Sample Size

In Section 3.3.1 a sample size of 500 was selected without justification. It was necessary to delay the justification of sample size until after a discussion of the analysis techniques.

Sample size can be determined directly from tables provided in Kraemer and Thiemann (1987). To use the tables, four pieces of information are needed: the significance level, the power, the effect size, and the statistical technique. The significance level, which is the probability of incorrectly rejecting an hypothesis, was set at 0.05, which is considered a standard in MIS research (Baroudi and Orlikowski, 1989). Power is the probability of correctly detecting a relationship, if one exists. A widely accepted norm for power is 80 percent (Baroudi and Orlikowski, 1989). Effect size represents the strength of the relationship among the variables in a population (Baroudi and Orlikowski, 1989). In new areas of research, such as MIS, effect sizes are likely to be small because the phenomena under study are typically not under good experimental or measurement control or both (Cohen, 1969). If the effect is thought to be small, an effect size of 0.2 or 0.3 should be used (Cohen, 1969).

Based upon the previous discussion, the following values are assigned to the needed parameters: significance = 0.05; power = 0.80; effect size = 0.30. The primary types of analysis used in this study are t-test for means and contingency tables. For a t-test, with the aforementioned parameter values, a sample size of 85 is needed (Kraemer and Thiemann, 1987). For a contingency table with one degree of freedom (i.e., 2 x 2), a sample size of 82 is needed (Kraemer and Thiemann, 1987).

According to the sample size calculations, the required sample sizes range from 82 to 85. It is not unrealistic to expect an 18 percent response rate in this study because Mahmood (1987) obtained a response rate of 18 percent using a similar data collection technique. If 18 percent of the 500 sampled organizations respond with only one project, then a usable sample of 90 will be obtained. Of course, if organizations provide two projects as asked, an 18 percent response rate will provide 180 samples - more than enough needed for this study. Therefore, to obtain a sample of at least 85, it is necessary to mail questionnaires to 500 organizations.

3.5 Chapter III Summary

This third chapter has first presented the propositions and hypotheses that will be examined in this study. One set of hypotheses is related to the selection of a prototyping strategy based upon characteristics of the project, user, developer, and organization. The second set of hypotheses looks at the influence on system success by the properly chosen prototyping strategy.

The research methodology used in this study was also presented in this chapter. The sample and data collection methods were discussed at length, and the independent and independent variables were reviewed.

The next part of the chapter examined the measurement of the model variables. To accomplish this, the questions used in the questionnaire were provided and explained.

The final chapter section looked at possible data analysis techniques that will be used. This section only considered some of the more likely analysis methods.

CHAPTER IV

RESULTS

4.1 Overview

This chapter describes the results of tests of hypotheses proposed in Chapter III. The interpretation and implications of the results will be covered in the next chapter. This chapter begins with an examination of the validity and reliability of the questionnaire which was used to collect the data. Next, a discussion of response rate and sample representativeness is provided. Then, sample characteristics, in the form of descriptive statistics, are discussed. The remaining segment of the chapter is dedicated to presenting the results of the various hypothesis tests.

4.2 Instrument Validity

The validity of a questionnaire is the extent to which the questionnaire actually measures what it is supposed to measure (Bailey and Pearson, 1983; Baroudi and Orlikowski, 1988). In this study, the only perceptions measured are those used to determine system success vis-a-vis user information satisfaction. To determine the validity of the UIS measure, construct and convergent validity are examined.

4.2.1 Construct Validity

Construct validity indicates the degree to which an operationalization of a variable actually captures the concept it purports to measure (Baroudi and Orlikowski, 1988; Sethi and King, 1991). Two methods of construct validation have been suggested (Nunnally, 1978): (1) correlation analysis; and (2) factor analysis.

4.2.1.1 Correlation Analysis

Correlation analysis is used to measure the degree of association between each item of the instrument and the total score (Baroudi and Orlikowski, 1988; Nunnally, 1978; Sethi and King, 1991). Each item is considered construct valid to the extent it correlates positively with the total score.

Each item's score has been removed from the total score to control for spurious part-whole correlations. The results of the correlation analysis are shown in Table 14. All correlations are significant at $p \leq 0.0001$. Based upon the correlation analysis, the instrument has construct validity.

4.2.1.2 Factor Analysis

Factor analysis is a technique of finding clusters of related variables (Nunnally, 1978). Each cluster is denoted by a group of variables whose members correlate more highly among themselves than they do with variables not included in the cluster. For construct validity, factor analysis indicates that a group of items measure the same things. The four questions used to measure UIS in this study should load on one factor, as demonstrated in other studies (e.g., Baroudi and Orlikowski, 1988; Ives, Olson, and Baroudi, 1983).

TABLE 14

ITEM CORRELATIONS WITH TOTAL SCORE

Item	Correlation with Total Score
Relevancy of output	0.65
Accuracy of output	0.84
Reliability of output	0.90
Precision of output	0.84

* All correlations significant at $p \leq 0.0001$

A principal components analysis was used as the extraction technique and VARIMAX as the method of rotation. The number of factors were not a priori specified and a minimum eigenvalue of one was required.

As suspected, only one factor emerged. The factor loadings are provided in Table 15. As seen, all items load very high on the single factor. Therefore, it can be concluded that these four items are measuring the same construct.

TABLE 15

FACTOR LOADINGS

Item	Factor Loading
Relevancy of output	0.67
Accuracy of output	0.89
Reliability of output	0.97
Precision of output	0.88

4.2.2 Convergent Validity

Convergent validity is the extent to which a measure is correlated or agrees with other measures of the same construct (Baroudi and Orlikowski, 1988; Sethi and King, 1991). In this case, the measures of UIS are correlated with a single question that asked: "How would you rate your satisfaction with this system?" The results are presented in Table 16. All correlations are significant at $p \leq 0.0001$.

TABLE 16

UIS CORRELATIONS WITH OVERALL MEASURE

Item	Correlation with Overall Measure
Relevancy of output	0.68
Accuracy of output	0.72
Reliability of output	0.81
Precision of output	0.74

* All correlations significant at $p \leq 0.0001$

4.3 Instrument Reliability

After determining the validity of an instrument, it is necessary to assess the reliability of the instrument. Reliability is the extent to which the questionnaire is free from measurement error (Baroudi and Orlikowski, 1988). High correlations between alternative measures or large Cronbach's alphas are usually signs that the measures are reliable (Straub, 1989).

Table 16 shows the correlation between each item and an overall measure which can be used to partially demonstrate reliability (when the overall measures is considered an alternative measure). However, Cronbach's alpha provides a better measure of reliability.

The Cronbach's alphas for the four-item UIS instrument are provided in Table 17. With the exception of the "relevancy of output" question, all of the items have an alpha greater than 0.80. A Cronbach's alpha of 0.80 or higher is considered appropriate, although 0.60 is acceptable in exploratory research (Nunnally, 1978). The overall Cronbach's alpha for the four items collectively is 0.91. Thus, this instrument seems to demonstrate high reliability.

TABLE 17

CRONBACH'S ALPHAS

Item	Cronbach's Alpha
Relevancy of output	0.65
Accuracy of output	0.84
Reliability of output	0.90
Precision of output	0.88

* Overall Cronbach's Alpha = 0.91

4.4 Response Rate

The results described in this chapter are based on data obtained via a mail survey conducted in the Spring of 1993. The survey instrument was mailed to 500 systematically selected organizations obtained from the Spring 1992 edition of the Directory of Top Computer Executives. Twenty-two of the questionnaires were returned because of bad addresses. Each "bad address" was replaced by a randomly selected organization so that a sample of 500 could be maintained. The entire survey procedure is described in more detail in Chapter 3.

A total of 119 companies responded to the survey (24%). Unfortunately, 50 responses from 45 different organizations were deemed unusable. Reasons a response could not be used include: both the user questionnaire and developer questionnaire were not returned; or one of the questionnaires did not contain sufficient information to be usable. However, responses from 74 of the companies could be used (15%). Of the 74 companies that responded, 49 provided data for one project, 24 provided data for two different projects, and one provided data for three projects, for a total of 100 projects (i.e., n=100). The response rate is summarized in Table 18.

Although the response rate is somewhat low, it was not unexpected. Several factors contributed to the low response rate:

1. The questionnaires were fairly lengthy.
2. Both the developer and user questionnaires needed to be returned to be useful.
3. Many companies have a policy against completing unsolicited questionnaires.
4. Many companies have not developed new systems in the past two years (as specified in this study).

TABLE 18

RESPONSE RATE

	Responses		
	Usable	Unusable	Total
COMPANIES Number of companies responding	74	45	119
PROJECTS Number of companies reporting..			
1 project	49	40	89
2 projects	24	5	29
3 projects	1	0	1
Total Projects Reported	100	50	150

As determined in Chapter 3, a sample of 85 projects was needed to provide sufficient power for the statistical tests performed herein. Obviously, the 100 projects obtained in this survey is sufficiently large to provide strong power to the tests.

4.5 Representativeness of the Sample

Although a sufficiently large sample size was obtained (i.e., 100), the sample is virtually "worthless" if it is not representative of the population for which we are trying to study (West, 1963). To verify the representativeness of the sample, two different approaches are used. First, the representativeness is tested by comparing the industries of the respondents and population. Second, as suggested by Oppenheim (1966) and West (1963), the respondents are compared by time of response using various demographics.

4.5.1 Representativeness by Industry

Table 19 provides information on the population (from the Spring 1992 Directory of Top Computer Executives), the sample, and the respondents. In comparing the sample against the population (column 2 of Table 19), we find that the only significant difference occurs in the "Other" category (p=0.04); all other differences between the population and selected sample are not significant. This demonstrates that the selected sample adequately represented the population.

A more important comparison is between the population and the respondents. As seen in column 3 of Table 19, there are no significant differences between the population and the respondents based on industry. Therefore, it can be concluded that, based upon industry, the respondents properly represent the population.

4.5.2 Representativeness by Time of Response

Representativeness is also evaluated by comparing the first set of responses against the last set of responses. The "first set of responses" is comprised of those projects that were received as a result of the first mailing of the questionnaires. The "last set of responses" is comprised of those projects that were received as a result of the follow-up questionnaire. Roughly, these can be divided into the responses received the first five weeks (first set) and responses received the last three weeks (last set) of an eight week data collection phase. According to Oppenheim (1966) and West (1963, 1991), those respondents who have to be reminded to respond, via follow-ups, are taken to typify those who do not respond at all. The "first set" contains 65 projects; the "second set" contains 35 projects.

TABLE 19

SAMPLE REPRESENTATIVENESS BY INDUSTRY

Industry	Population n=13,693	Sample n=500	Respondents n=74
Manufacturing/Service	52.3%	51.1% (p=0.58)	45.9% (p=0.53)
Financial	7.3%	7.7% (p=0.73)	2.7% (p=0.13)
Insurance	5.0%	6.5% (p=0.15)	4.1% (p=0.71)
Retail	4.7%	5.4% (p=0.46)	9.5% (p=0.16)
Transportation	1.5%	1.7% (p=0.71)	4.1% (p=0.40)
Education	8.4%	6.9% (p=0.23)	6.8% (p=0.61)
Health Service	4.9%	4.6% (p=0.76)	4.1% (p=0.86)
Government	12.1%	10.5% (p=0.27)	16.2% (p=0.28)
Other	3.8%	5.6% (p=0.04)	6.8% (p=0.18)

To check the representativeness based upon the time of response, the respondents are examined along the following dimensions: size of company as measured by number of employees, number of information systems (IS) personnel, and whether or not prototyping was used (i.e., use of prototyping). As shown in Table 20, there are no significant differences between the "first set" and "last set" of respondents. Once again, it can be concluded that the respondents are representative of the population.

TABLE 20

SAMPLE REPRESENTATIVENESS
BASED ON TIME OF RESPONSE

	1st Set of Respondents (n=65)	2nd Set of Respondents (n=35)	p-value
Number of employees	Avg = 2568	Avg = 2613	p=0.97
Number of IS personnel	Avg = 61	Avg = 69	p=0.77
Use of Prototyping	Avg = 1.23	Avg = 1.23	p=0.98

4.6 Sample Characteristics

As shown earlier in Table 19, the sample represents a wide variety of industries. Additional demographics are illustrated in Table 21 and discussed below.

4.6.1 Respondents

Several types of developers responded to the survey. Developer categories were determined by a fill-in-the-blank question from the questionnaire that asked the developer to specify his/her title. The largest group of developer respondents are programmer/analysts, although all groups are represented well. Those included in the "Other" category have such job titles as "Computer Specialist" and "Information Specialist." Developer respondents have an average (mean) work experience of 16 years.

User respondents were categorized as either management, staff/clerical, or professional. An example from each would include Production Supervisor, Payroll Clerk, and Nurse, for management, staff/clerical, and professional, respectively. All groups were well represented, with Management commanding the largest segment with 51 percent.

4.6.2 Organizations

A diverse set of organizations, based on size, is represented by this sample. Size is measured here by the number of company employees, the number of IS personnel, and the IS budget. The average organization in this sample has 2584 employees, although the range is from seven employees to 51,000 employees. For the information systems department, the average number of personnel is 64, with a range of one IS person to 800 IS personnel. The average budget of the IS department is $12,456,236. The budget range is from $95,000 to $300,000,000. As demonstrated, the sample represents a wide variety of firms, not only by industry, but also by size.

4.6.3 Projects

Of the projects reported in this study, 73 percent used prototyping (as defined in Chapter 1), and 27 percent did not use prototyping. The use of prototyping found in this study is somewhat higher than usage reported in other studies (e.g., Doke (1990) found a 60 percent usage rate).

Other noteworthy project characteristics include the size of the development team, the time needed to develop the system (in months and man-hours), and the development cost. The development team size ranges from one person to 25, with an average team size of five. The average development time, in months, is 14; the average development man-hours is 19,397. The months range from a low of one month to a high of seven years (i.e., 84 months). Man-hours range from 40 to over one million (1,152,000). The cost of the average project is $396,500 with a range of $4,355 to $4,000,000. As these project characteristics demonstrate, the sample represents a variety of systems.

Another important characteristic is the measurement of system success via the UIS instrument. As seen, the average "success" on a scale of -12 to +12 is 7. Obviously, the measure is skewed, and can potentially limit the findings in the study.

TABLE 21

SAMPLE DEMOGRAPHICS

Respondents

Developer		User	
Systems Analyst	11%	Management	51%
Programmer	6%	Staff/Clerical	33%
Programmer/Analyst	26%	Professional	16%
Manager	22%		
Project Leader	15%		
MIS Director	15%		
Other	5%		
Years Experience: Mean=16 Range = 2 to 33			

Organizations

Characteristic	Mean	Median	Minimum	Maximum
Number of Employees	2584	550	7	51,000
Number of IS Staff	64	20	1	800
IS Budget	$12,456,236	$2,250,000	$95,000	$300,000,000

Projects

Characteristic	Mean	Median	Minimum	Maximum
Team Size	5	4	1	25
Months to Develop	14	12	1	84
Manhours to Develop	19,397	2000	40	1,152,000
Development Cost	$396,500	$110,000	$4,355	$4,000,000
Success	7	8	-11.5	12
Use of Prototyping	YES = 73%	NO = 27%		

4.7 Hypotheses Testing

The next sections, 4.7.1 through 4.7.18, examine each of the 18 contingency hypotheses investigated in this study, based upon the data obtained from the survey. Hypothesis 19, which is an examination of system success based upon the "proper" selection of a prototyping strategy, will be tested as a subset of each of the 18 contingency hypotheses and is also presented in Sections 4.7.1 through 4.7.18.

4.7.1 Hypothesis 1

Hypothesis 1: If requirements are unclear, then prototyping will be used.

A single question was used to measure the clarity of the requirements. A response range of one to seven is possible, where one is equivalent to "requirements specified completely" and seven is "requirements not specified." Thus, to test this hypothesis, a t-test for the mean response to the question is used. Table 22 illustrates the results of the t-test.

TABLE 22

HYPOTHESIS 1

Prototyping?	N	Mean	Prob>|T|
NO	26	3.19	0.97
YES	73	3.18	

It is hypothesized that the larger responses would be associated with prototyping, and the lower responses associated with "no" prototyping. Instead, as shown in Table 22, there appears to be no relationship between the use of prototyping and the clarity of requirements.

If the guideline suggested by Hypothesis 1 is followed, would it make a difference in the success of the system? Essentially, this is the question that Hypothesis 19 is investigating. Success, a measurement of the user's satisfaction, is determined by a score that ranges from -12 to +12 (as specified in Chapter 3). To determine if a project is in agreement with the hypothesis, the following decision rule is used:

```
If clarity response >= 4 and prototyping is used,
then the data agrees with Hypothesis 1,
else if prototyping is not used,
     then the data does not agree with Hypothesis 1.
If clarity response < 4 and prototyping is not used,
then the data agrees with Hypothesis 1,
else if prototyping is used,
     then the data does not agree with Hypothesis 1.
```

As indicated earlier in this section, a seven-point scale is used for measuring clarity of requirements. Therefore, the mid-point of "4" is used as a way to dichotomize the response into "agreement/no agreement" such that the decision rule specified above could be constructed. During the remaining course of the hypothesis discussions, the mid-point of "4" will be used to dichotomize the response - if a seven-point scale is used. For continuous valued scales, the mean or median will be used.

TABLE 23

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 1

Correct?	N	Mean	Prob>|T|
NO	51	6.98	0.99
YES	43	6.99	

As shown in Table 23, although the system success is negligibly higher when the guideline is followed, the difference is not significant. Therefore, Hypothesis 19, as determined by Hypothesis 1, is not supported.

4.7.2 Hypothesis 2

Hypothesis 2: If the system requirements are dynamic, then prototyping will be used.

The data for Hypothesis 2 is obtained from a single question concerning the stability of requirements. A seven-point scale, ranging from one ("no changes") to seven ("many changes"), was used. It is suspected that the smaller responses to the seven-point scale will be associated with "no" prototyping, and the larger responses will be associated with the use of prototyping. A t-test for means is used to investigate this hypothesis, as shown in Table 24.

TABLE 24

HYPOTHESIS 2

Prototyping?	N	Mean	Prob>|T|
NO	27	4.56	0.67
YES	73	4.70	

Although the responses associated with the use of prototyping (i.e., YES) have a higher mean, the difference is not significant. Thus, we cannot conclude that stability of requirements is related to the selection of a prototyping strategy.

To test Hypothesis 19, as determined by Hypothesis 2, we use a decision rule similar to the one given for Hypothesis 1:

```
If stability response >= 4 and prototyping is used,
then the data agrees with Hypothesis 2,
else if prototyping is not used,
     then the data does not agree with Hypothesis 2.
If stability response < 4 and prototyping is not used,
then the data agrees with Hypothesis 2,
else if prototyping is used,
     then the data does not agree with Hypothesis 2.
```

Although the success is higher when the guideline suggested by Hypothesis 2 is followed, the difference is not significant (see Table 25). Thus, compliance with Hypothesis 2 does not affect system success.

TABLE 25

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 2

Correct?	N	Mean	Prob>|T|
NO	33	6.77	0.73
YES	62	7.13	

4.7.3 Hypothesis 3

Hypothesis 3: If the system mode is on-line, then prototyping will be used.

System mode is classified as on-line, batch, or elements of both, as determined by a single question on the questionnaire. Systems that were classified as having "elements of both" were eliminated from the analysis because of the specific statement of the hypothesis.

Due to the dichotomous nature of the response (i.e., Batch/On-line), a 2x2 contingency table is used to test Hypothesis 3; the appropriate statistic is the Chi-square statistic. The results are shown in Table 26.

As seen in Table 26, very few of the systems are classified as either (strictly) "batch" or "on-line." As a consequence, the small number of counts in the cells may invalidate the Chi-square statistic. Therefore, the p-value given in Table 26 must be interpreted with these facts in mind.

TABLE 26

HYPOTHESIS 3

		Prototyping	
		No	Yes
MODE	Batch	2*	1
	On-Line	8	21*

Chi-square = 1.93
p-value = 0.16
* = correct choice

Also shown in Table 26 are indications of the "correct" choices, based upon the statement of Hypothesis 3. Note that 23 of the 32 reported projects followed the suggested guideline. To investigate Hypothesis 19, we need to know if compliance with the guideline influences system success. Table 27 illustrates the resulting t-test which is used to test Hypothesis 19.

The t-test does not reveal a significant difference. However, this particular test will suffer from low statistical power due to the small sample used, which means a significant difference may actually exist but was undetectable because of low power.

TABLE 27

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 3

Correct?	N	Mean	Prob>|T|
NO	9	6.44	0.96
YES	21	6.36	

4.7.4 Hypothesis 4

Hypothesis 4: If the duration is long, then prototyping will not be used.

An open-ended question, asking the respondent to provide the number of months needed to develop the system, supplies the information necessary to test this hypothesis. A t-test of the mean months comparing those projects utilizing prototyping to those not utilizing prototyping is used, as shown in Table 28.

TABLE 28

HYPOTHESIS 4

Prototyping?	N	Mean	Prob>|T|
NO	24	11.56	0.19
YES	71	15.15	

The data suggests that prototyping is used for longer duration projects, which is opposite of the hypothesized situation. However, nothing definite can be concluded because the difference is not statistically significant.

To examine the influence on system success of a prototyping strategy chosen in accordance with Hypothesis 4, it is necessary to use a measure of central tendency as a way of dichotomizing the data. For this particular response, the average is 14, and the median is 12. Because the data is not normally distributed due to outliers at the high-end, the median of 12 is used. Therefore, the decision rule, based upon a median duration of 12 months, is:

```
If duration >= 12 months and prototyping is not used,
then the data agrees with Hypothesis 4,
else if prototyping is used,
     then the data does not agree with Hypothesis 4.
If duration < 12 months and prototyping is used,
then the data agrees with Hypothesis 4,
else if prototyping is not used,
     then the data does not agree with Hypothesis 4.
```

TABLE 29

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 4

Correct?	N	Mean	Prob>|T|
NO	49	6.80	0.51
YES	42	7.44	

As seen in Table 29, projects that followed the guideline of Hypothesis 4 had a higher level of success, but the difference is not significant (p = 0.51).

4.7.5 Hypothesis 5

Hypothesis 5: If innovation is high, then prototyping will be used.

Innovation is measured on a seven-point scale, anchored by "new development" and "minor modification to existing system." It is hypothesized that new developments (high innovation) will be associated with the use of prototyping, and minor modifications (low innovations) will be associated with "no" prototyping. A t-test is used to investigate this question. The data is presented in Table 30.

TABLE 30

HYPOTHESIS 5

Prototyping?	N	Mean	Prob>|T|
NO	25	2.52	0.11
YES	73	2.00	

The data suggests that the use of prototyping is associated with higher innovations. However, the lack of significant differences will not allow a definite conclusion.

If Hypothesis 5 is followed, what is the effect on system success? The following decision rule is used to categorize the data:

```
If innovation <= 4 and prototyping is used,
then the data agrees with Hypothesis 5,
else if prototyping is not used,
     then the data does not agree with Hypothesis 5.
If innovation > 4 and prototyping is not used,
then the data agrees with Hypothesis 5,
else if prototyping is used,
     then the data does not agree with Hypothesis 5.
```

As illustrated in Table 31, higher success is associated with the compliance of Hypothesis 5. However, the difference is not significant.

TABLE 31

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 5

Correct?	N	Mean	Prob>|T|
NO	30	5.87	0.14
YES	63	7.56	

4.7.6 Hypothesis 6

Hypothesis 6: If project size is large, then prototyping will be used.

Project size is determined by three different measures: (1) the project budget as a percentage of the department budget (PER-BUD); (2) the man-hours needed to build the system; and (3) the rank, in development cost, of the system compared to other systems developed by the department.

For the PER-BUD variable, larger sized projects are indicated by the larger PER-BUD numbers. Hypothesis 6 suggests that larger projects should utilize prototyping. A t-test using PER-BUD is used as one test of this hypothesis. The results are shown in Table 32.

TABLE 32

HYPOTHESIS 6 (PER-BUD)

Prototyping?	N	Mean	Prob>|T|
NO	6	0.42	0.32
YES	28	0.16	

As indicated in Table 32, the larger PER-BUD is associated with "no" prototyping, which is opposite of the statement of hypothesis. However, two things must be noted. First, the difference is not significant, thus, a conclusion cannot be reached. Second, the total number of respondents to this question is only 34, which greatly reduces the statistical power. With low statistical power, we do not know if a relationship does not exist, or if the low power prevented us from detecting the relationship.

When using man-hours as a surrogate for system size, it is hypothesized that the larger man-hour projects should use prototyping. A t-test is used to test this hypothesis; the results are shown in Table 33. The mean man-hours for projects utilizing prototyping are higher than the man-hours for other projects. However, the difference is not significant (p = 0.44).

TABLE 33

HYPOTHESIS 6 (man-hours)

Prototyping?	N	Mean	Prob>|T|
NO	18	8298	0.44
YES	65	22471	

The third way to measure system size is via the rank, in cost, of the system compared to other system. Rank is measured on a seven-point scale with the following anchors: "upper 10%" and "lower 10%". Hypothesis 6 suggests that projects ranked in the upper percentages should use prototyping. A t-test is used to investigate this suggestion (see Table 34).

The information presented in Table 34 does not support Hypothesis 6. Thus, there appears to be no relationship between the rank of the project and the use of prototyping.

TABLE 34

HYPOTHESIS 6 (rank)

Prototyping?	N	Mean	Prob>|T|
NO	19	3.68	0.56
YES	68	3.41	

To investigate the influence on system success of the proper prototyping strategy (as suggested by Hypothesis 6), each of the three measures of size - percent-of-budget, man-hours and rank - are used.

Percent-of-budget (PER-BUD) is a continuous variable which ranges from 0.0004 to 3. Therefore, to examine the influence on system success of a prototyping strategy chosen in accordance with Hypothesis 6, based upon PER-BUD, it is necessary to use a measure of central tendency as a way of dichotomizing the data. For this particular response, the average is 0.20, and the median is 0.03. Because the data is not normally distributed due to outliers at the high-end, the median of 0.03 is used. Therefore, the decision rule, based upon a median PER-BUD of 0.03, is:

```
If PER-BUD >= 0.03 and prototyping is not used,
then the data agrees with Hypothesis 6,
else if prototyping is used,
     then the data does not agree with Hypothesis 6.
If PER-BUD < 0.03 and prototyping is used,
then the data agrees with Hypothesis 6,
else if prototyping is not used,
     then the data does not agree with Hypothesis 6.
```

As indicated in Table 35, the success is higher when the decision rule is NOT followed. However, the difference is not significant (p = 0.26). Once again, it must be noted that a relationship may exist but was undetectable due to the small sample size.

TABLE 35

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 6
(PER-BUD)

Correct?	N	Mean	Prob>|T|
NO	14	8.29	0.26
YES	17	6.71	

Man-hours is also a continuous variables which must be dichotomized according to the median or mean. The data is not normally distributed because a large outlier on the high-end, therefore, the median is used. The decision rule, according to man-hours (based on a median man-hours), to determine the proper prototyping strategy is:

```
If man-hours >= 2000 and prototyping is used,
then the data agrees with Hypothesis 6,
else if prototyping is not used,
     then the data does not agree with Hypothesis 6.
If man-hours < 2000 and prototyping is not used,
then the data agrees with Hypothesis 6,
else if prototyping is used,
     then the data does not agree with Hypothesis 6.
```

The results of the previous decision rule are provided in Table 36. Success is higher when the suggested contingency is NOT followed, however, the difference is not significant (p = 0.42).

TABLE 36

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 6
(man-hours)

Correct?	N	Mean	Prob>|T|
NO	38	7.44	0.42
YES	41	6.60	

The final measure of system size is rank (in cost). Rank is measured on a seven-point scale. Thus, the decision rule, according to rank, is:

If rank response < 4 and prototyping is used,
then the data agrees with Hypothesis 6,
else if prototyping is not used,
    then the data does not agree with Hypothesis 6.
If rank response >= 4 and prototyping is not used,
then the data agrees with Hypothesis 6,
else if prototyping is used,
    then the data does not agree with Hypothesis 6.

The results are provided in Table 37. Compliance with the previous decision rule does not appear to influence system success.

It appears that success is higher when prototyping is used for smaller projects (as measured by PER-BUD and man-hours), but a definite conclusion cannot be reached because of statistical insignificance. Therefore, the only conclusion that can be reached is: it does not seem that system success is influenced by the suggested guideline of Hypothesis 6 as measured by PER-BUD, man-hours, and rank.

TABLE 37

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 6
(rank)

<table>
<tr><th>Correct?</th><th>N</th><th>Mean</th><th>Prob>|T|</th></tr>
<tr><td>NO</td><td>42</td><td>6.85</td><td>0.94</td></tr>
<tr><td>YES</td><td>40</td><td>6.76</td><td></td></tr>
</table>

4.7.7 Hypothesis 7

Hypothesis 7: If the system is a critical system, then prototyping will be used.

A critical system is a system that operates, manages, and controls the daily business activities of the organization. A single question, with possible responses of either "yes" or "no", was used to determine whether the system is critical or not. Due to the dichotomous nature of the response (i.e., YES/NO), a 2x2 contingency table is used to test Hypothesis 7; the appropriate statistic is the Chi-square statistic. The results are shown in Table 38.

TABLE 38

HYPOTHESIS 7

	Prototyping	
	No	Yes
Critical	20	43*
Non-Critical	6*	30

Chi-square = 2.69
p-value = 0.10
* = correct choice

The small p-value (p=0.10) indicates a great deal of dependence between the prototyping strategy and whether or not the system is critical. However, a Phi-coefficient of -0.165 indicates a negative correlation among the data, which leads us to conclude a relationship that is opposite of that hypothesized.

Also shown in Table 38 are indications of the "correct" choices, based upon the statement of Hypothesis 7. To investigate Hypothesis 19, we need to know if compliance with the guideline influences system success. Table 39 illustrates the resulting t-test which is used to test Hypothesis 19. As seen, system success is not influenced by the selection of a prototyping strategy according to Hypothesis 7.

TABLE 39

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 7

Correct?	N	Mean	Prob>|T|
NO	48	7.15	0.74
YES	46	6.82	

4.7.8 Hypothesis 8

Hypothesis 8: If a system has stringent performance requirements, then an evolutionary prototyping strategy will not be used.

To determine the performance requirements, a single question with the anchors "strict requirements" to "low requirements" is used. Hypothesis 8 suggests that an evolutionary prototyping strategy should not be used for strict requirements. A t-test is used to examine this hypothesis. According to the information provided in Table 40, Hypothesis 8 is not supported.

TABLE 40

HYPOTHESIS 8

Prototyping?	N	Mean	Prob>|T|
Evolutionary	19	3.11	0.96
Other	81	3.12	

If the guideline suggested by Hypothesis 8 is followed, would system success be affected? The following decision rule is be used to determine compliance by a project:

If response < 4 and prototyping strategy is "other",
then the data agrees with Hypothesis 8,
else if evolutionary prototyping is used,

then the data does not agree with Hypothesis 8.
If response >= 4 and evolutionary prototyping is used,
then the data agrees with Hypothesis 8,
else if the prototyping strategy is "other",
then the data does not agree with Hypothesis 8.

As demonstrated in Table 41, system success is significantly higher for those projects that followed the guideline suggested by Hypothesis 8.

TABLE 41

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 8

Correct?	N	Mean	Prob>|T|
NO	43	5.90	0.04
YES	52	7.92	

4.7.9 Hypothesis 9

Hypothesis 9: If the users do not have time to dedicate to the project, then prototyping will not be used.

A single question with a seven-point scale is used to determine the user's participation in the development process. With the anchors of "participated to a great extent" and "did not participate," it is hypothesized that prototyping should not be used when users do not participate in development. A t-test is used to test Hypothesis 9.

TABLE 42

HYPOTHESIS 9

Prototyping?	N	Mean	Prob>¦T¦
NO	27	3.56	0.003
YES	68	2.38	

Strong evidence supports Hypothesis 9, as shown in Table 42. As suspected, higher user participation is associated with the use of prototyping. Since Hypothesis 9 is supported, it is necessary to investigate the effect on system success (i.e., Hypothesis 19). The test for the effect on system success is shown in Table 43. The decision rule to determine compliance with Hypothesis 9 is:

If response <= 4 and prototyping is used,
then the data agrees with Hypothesis 9,
else if prototyping is not used,
    then the data does not agree with Hypothesis 9.
If response > 4 and prototyping is not used,
then the data agrees with Hypothesis 9,
else if prototyping is used,
    then the data does not agree with Hypothesis 9.

We observe from Table 43, that, although Hypothesis 9 is supported, increased levels of user participation coupled with the use of prototyping have no significant effect on system success. Interestingly, system success is lower when the hypothesis is followed (although, once again, it is not significant).

TABLE 43

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 9

Correct?	N	Mean	Prob>|T|
NO	25	7.40	0.56
YES	70	6.86	

4.7.10 Hypothesis 10

Hypothesis 10: If the users are inexperienced with prototyping, then prototyping will not be used.

A user's prior experience with prototyping is indicated by his/her response to a single question with the following anchors: "no knowledge" to "extensive knowledge." It is suspected that higher levels of prior prototyping knowledge will be associated with the usage of prototyping. A t-test of the mean response is used to test this hypothesis, as shown in Table 44.

Evidence suggests that Hypothesis 11 should not be rejected. We can conclude that a relationship between the use of prototyping and a user's prior knowledge of prototyping exists.

TABLE 44

HYPOTHESIS 10

Prototyping?	N	Mean	Prob>¦T¦
NO	27	2.48	0.005
YES	68	3.68	

Since this guideline (Hypothesis 10) is followed in practice, is system success influenced by its compliance? The test for the effect on system success is shown in Table 45. The decision rule to determine compliance with Hypothesis 10 is:

If response >= 4 and prototyping is used,
then the data agrees with Hypothesis 10,
else if prototyping is not used,
    then the data does not agree with Hypothesis 10.
If response < 4 and prototyping is not used,
then the data agrees with Hypothesis 10,
else if prototyping is used,
    then the data does not agree with Hypothesis 10.

TABLE 45

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 10

Correct?	N	Mean	Prob>|T|
NO	33	7.64	0.28
YES	62	6.70	

Although Hypothesis 10 is supported, its compliance has no significant effect on system success (see Table 45). Interestingly, system success is lower when the guideline is followed (although, once again, it is not significant).

4.7.11 Hypothesis 11

Hypothesis 11: If the users have no automation experience, then prototyping will be used.

To determine a user's prior experience with automation, a single question with the anchors "no experience" to "extensive experience" is used. Hypothesis 11 suggests that prototyping should be used when users have no automation experience. A t-test is used to examine this hypothesis (see Table 46).

As seen in Table 46, the mean experience rating of the users associated with prototyping is higher than the converse, but not significantly. Therefore, Hypothesis 11 is not supported.

TABLE 46

HYPOTHESIS 11

Prototyping?	N	Mean	Prob>|T|
NO	27	5.04	0.19
YES	67	5.49	

Would system success be affected if the guideline suggested by Hypothesis 11 is followed? The following decision rule is be used to determine compliance:

If response < 4 and prototyping is used,
then the data agrees with Hypothesis 11,
else if prototyping is not used,
then the data does not agree with Hypothesis 11.
If response >= 4 and prototyping is not used,
then the data agrees with Hypothesis 11,
else if prototyping is used,
then the data does not agree with Hypothesis 11.

TABLE 47

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 11

Correct?	N	Mean	Prob>|T|
NO	65	7.57	0.12
YES	29	5.66	

It is interesting to note that success is higher when the guideline is not followed (see Table 47). However, the influence on system success is not significant, thus, a definite conclusion cannot be drawn.

4.7.12 Hypothesis 12

Hypothesis 12: If there are a large number of users, then prototyping will not be used.

An open-ended question, asking the respondent to provide the number of users that interacted with developers during system development, supplies the information necessary to test this hypothesis. A t-test of the mean number of users comparing those projects utilizing prototyping with those not utilizing prototyping is used, as shown in Table 48.

The data suggests that no relationship exists between the number of users interacting with developers and the use of prototyping.

TABLE 48

HYPOTHESIS 12

Prototyping?	N	Mean	Prob>|T|
NO	25	9.68	0.98
YES	73	9.77	

To examine the influence on system success of a prototyping strategy chosen in accordance with Hypothesis 12, it is necessary to use a measure of central tendency because of the continuous nature of the measure. The measure is normally distributed, thus the average (mean) number of users provides a good indicator for dichotomizing the data. The decision rule, based upon an average number of users of 9.75, is:

> If number of users > 9.75 and prototyping is not used,
> then the data agrees with Hypothesis 12,
> else if prototyping is used,
>   then the data does not agree with Hypothesis 12.
> If number of users < 9.75 and prototyping is used,
> then the data agrees with Hypothesis 12,
> else if prototyping is not used,
>   then the data does not agree with Hypothesis 12.

As shown in Table 49, system success is significantly higher when the guideline suggested by Hypothesis 12 is followed.

TABLE 49

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 12

Correct?	N	Mean	Prob>|T|
NO	36	5.77	0.08
YES	57	7.72	

4.7.13 Hypothesis 13

Hypothesis 13: If the user impact is high, then prototyping will be used.

Two questions, both with seven-point scales, are used to measure the user impact. The responses from the two questions are averaged to determine the user impact. It is suspected that for higher levels of user impact, prototyping should be used. A t-test is used to investigate this hypothesis (see Table 50).

TABLE 50

HYPOTHESIS 13

Prototyping?	N	Mean	Prob>|T|
NO	27	3.78	0.89
YES	68	3.74	

There appears to be no relationship between the user impact and the use of prototyping. However, if Hypothesis 13 is followed, would it affect system success? This question can be answered by categorizing the projects according to the following decision rule, and then using a t-test on the success between those projects that correctly follow the guideline and those that do not:

> If impact response < 4 and prototyping is not used,
> then the data agrees with Hypothesis 13,
> else if prototyping is used,
>   then the data does not agree with Hypothesis 13.
> If impact response >= 4 and prototyping is used,
> then the data agrees with Hypothesis 13,
> else if prototyping is not used,
>   then the data does not agree with Hypothesis 13.

TABLE 51

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 13

Correct?	N	Mean	Prob>|T|
NO	49	7.54	0.25
YES	46	6.43	

Based upon the results provided in Table 51, it does not appear that following the guideline of Hypothesis 13 affects system success.

4.7.14 Hypothesis 14

Hypothesis 14: If developers have experience with similar applications, then prototyping will not be used.

A single seven-point scale question with the anchors "no experience" and "extensive experience" is used to measure a developer's prior experience with similar applications. It is hypothesized that higher levels of experience would not require the use of prototyping. A t-test of the response is used to test this hypothesis, as shown in Table 52.

TABLE 52

HYPOTHESIS 14

Prototyping?	N	Mean	Prob>|T|
NO	26	5.27	0.10
YES	72	4.58	

The data presented in Table 52 provides support for Hypothesis 14 (at p=0.10). We can conclude that the use of prototyping is related to a developer's application experience. Since a relationship does exist, it is necessary to determine the extent on system success of compliance with the guideline. These results are shown in Table 53. The following decision rule is used to determine the categories of "correct" or "not correct" according to Hypothesis 14:

If response <= 4 and prototyping is not used,
then the data agrees with Hypothesis 14,
else if prototyping is used,

then the data does not agree with Hypothesis 14.
If response > 4 and prototyping is used,
then the data agrees with Hypothesis 14,
else if prototyping is not used,
then the data does not agree with Hypothesis 14.

Although a significant relationship exists between developer experience and prototyping strategy, it does not appear to affect system success (see Table 53). It should be noted that the mean success is higher when the guideline is followed, but the difference is not significant.

TABLE 53

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 14

Correct?	N	Mean	Prob>|T|
NO	46	6.23	0.14
YES	47	7.70	

4.7.15 Hypothesis 15

Hypothesis 15: If developers are not experienced with prototyping, then prototyping will not be used.

Hypothesis 15 states that developers lacking previous experience with prototyping should not use prototyping. A single question with the anchors "no experience" and "extensive experience" is used to measure prototyping experience. Thus, higher levels of experience should be associated with the use of prototyping. The t-test used to examine this hypothesis is illustrated in Table 54.

TABLE 54

HYPOTHESIS 15

Prototyping?	N	Mean	Prob>|T|
NO	25	2.88	0.0001
YES	72	5.36	

A significant difference exists between the mean responses according to a developer's prior prototyping experience and whether prototyping was used or not. Therefore, we can conclude that a relationship exists between the use of prototyping and a developer's prior experience with prototyping.

Is system success affected by following the suggestion of Hypothesis 15? The following decision rule is used to delineate the projects:

```
If response < 4 and prototyping is not used,
then the data agrees with Hypothesis 15,
else if prototyping is used,
     then the data does not agree with Hypothesis 15.
If response >= 4 and prototyping is used,
```

then the data agrees with Hypothesis 15,
else if prototyping is not used,
then the data does not agree with Hypothesis 15.

As seen in Table 55, system success is higher for those projects following the guideline (7.40 versus 4.11). However, this difference is not large enough to be considered statistically significant (p=0.15).

TABLE 55

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 15

Correct?	N	Mean	Prob>|T|
NO	13	4.11	0.15
YES	80	7.40	

4.7.16 Hypothesis 16

Hypothesis 16: If a project does not have management support, then prototyping will not be used.

Two questions, both with seven-point scales, are used to measure management support. The responses from the two questions are averaged to determine the degree of management support. It is suspected that for lower levels of management support, prototyping should not be used. A t-test is used to investigate this hypothesis (see Table 56). Based upon the available evidence, there does not appear to be a relationship between prototyping strategy and management support.

TABLE 56

HYPOTHESIS 16

Prototyping?	N	Mean	Prob>|T|
NO	26	5.38	0.85
YES	71	5.46	

The effect on system success can be examined by grouping the data according to the following decision rule:

If support response < 4 and prototyping is not used,
then the data agrees with Hypothesis 16,
else if prototyping is used,
then the data does not agree with Hypothesis 16.
If support response >= 4 and prototyping is used,
then the data agrees with Hypothesis 16,
else if prototyping is not used,
then the data does not agree with Hypothesis 16.

A t-test of the means of system success between projects following the guideline and those that do not is used to determine the effect of compliance with the guideline on system success (see Table 57). Compliance with Hypothesis 16 does not appear to influence system success.

TABLE 57

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 16

Correct?	N	Mean	Prob>|T|
NO	31	6.87	0.89
YES	62	7.02	

<u>4.7.17 Hypothesis 17</u>

Hypothesis 17: If a need for experimentation and learning before full commitment exists, then prototyping will be used.

A single question, with possible responses of either "yes" or "no", was used to determine whether the need for experimentation and learning existed. Due to the dichotomous nature of the response (i.e., YES/NO), a 2x2 contingency table is used to test Hypothesis 17; the appropriate statistic is the Chi-square statistic. The results are shown in Table 58.

The data presented in Table 58 suggests the lack of a relationship between prototyping strategy and the need for experimentation and learning.

Also shown in Table 58 are indications of the "correct" choices, based upon the statement of Hypothesis 17. To investigate Hypothesis 19, we need to know if compliance with the guideline influences system success. Table 59 illustrates the resulting t-test which is used to test Hypothesis 19. As seen, system success is not influenced by compliance with Hypothesis 17.

TABLE 58

HYPOTHESIS 17

	Prototyping	
	No	Yes
Experiment & Learning	9	33*
Not Needed	16*	39

Chi-square = 0.731
p-value = 0.393
* = correct choice

TABLE 59

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 17

Correct?	N	Mean	Prob>|T|
NO	44	7.00	0.88
YES	48	6.84	

4.7.18 Hypothesis 18

Hypothesis 18: If prototyping support tools are not available, then an evolutionary prototyping strategy will not be used.

Respondents were asked to classify the tools used to develop the system, according to the following categories: (1) no tools; (2) text editors/word processors; (3) DBMS; (4) 4GL; and (5) other. A response of 3, 4, or 5 (with sufficient explanation) indicated that proper prototyping tools were used. Conversely, a response of 1 or 2 indicated that non-prototyping tools were used. Therefore, the data was categorized as either having prototyping tools, or not having prototyping tools. Due to the dichotomous nature of the response (i.e., proper tools/no tools), a 2x2 contingency table is used to test Hypothesis 3; the appropriate statistic is the Chi-square statistic. The results are shown in Table 60.

The small p-value (p=0.017) indicates a great deal of dependence between the prototyping strategy (evolutionary or other) and whether or not tools are used. A Phi-coefficient of 0.238 indicates a positive correlation among the data, which allows us to conclude that the use of evolutionary prototyping is associated with the use of prototyping tools.

Also shown in Table 60 are indications of the "correct" choices, based upon the statement of Hypothesis 18. To investigate Hypothesis 19, we need to know if compliance with the guideline influences system success. Table 61 illustrates the resulting t-test which is used to test Hypothesis 19.

TABLE 60

HYPOTHESIS 18

TOOLS	Prototyping: Evol	Prototyping: Other
No	6	50*
Yes	13*	31

Chi-square = 5.677
p-value = 0.017
* = correct choice

TABLE 61

HYPOTHESIS 19 AS DETERMINED BY HYPOTHESIS 18

Correct?	N	Mean	Prob>|T|
NO	34	7.75	0.22
YES	61	6.59	

The t-test on system success does not reveal a significant difference (Table 61). Thus, although a relationship exists between evolutionary prototyping and the use of development tools, it does not appear to affect system success.

4.7.19 Summary of Hypothesis Testing

Table 62 summarizes the results of the various hypotheses tests. Of the 18 hypotheses (i.e., suggested contingencies), only five are followed in practice. Also, only two of the contingencies, if followed, would result in a significant increase in system success (Hypothesis 19). Of the five significant contingencies (column 2), none significantly affected system success.

Although it is disappointing that more hypotheses are not supported, it is not surprising. Almost all of the contingencies tested in this study (Hypotheses 1 through 18) have been introduced in the literature without testing. Unfortunately, the results here suggest that many of the suppositions concerning the use of prototyping are not true.

Hopefully, these results will help clarify some of the myths of prototyping. Also, although a contingency is followed, it does not appear that system success is affected (or at least, for the five significant ones found here). This contradicts many of the statements in the literature which proclaim many of the benefits of prototyping.

TABLE 62

SUMMARY OF HYPOTHESIS TESTING

Hypotheses	p-value	Hyp 19 p-value
H1: If requirements are unclear, then prototyping will be used.	0.97	0.99
H2: If the system requirements are dynamic, then prototyping will be used.	0.67	0.73
H3: If the system mode is on-line, then prototyping will be used.	0.16	0.96
H4: If the duration is long, then prototyping will not be used.	0.19	0.51
H5: If innovation is high, then prototyping will be used.	0.11	0.14
H6: If project size is large, then prototyping will be used. (PER-BUD, man-hours, rank)	0.32,0.44 0.56	0.26,0.41 0.94
H7: If the system is a critical system, then prototyping will be used.	0.10*(+)	0.74
H8: If a system has stringent performance requirements, then an evolutionary prototyping strategy will not be used.	0.96	0.04**
H9: If the users do not have time to dedicate to the project, then prototyping will not be used.	0.00***	0.56
H10: If the users are inexperienced with prototyping, then prototyping will not be used.	0.00***	0.28
H11: If the users have no automation experience, then prototyping will be used.	0.19	0.12
H12: If there are a large number of users, then prototyping will not be used.	0.98	0.08*
H13: If the user impact is high, then prototyping will be used.	0.89	0.25
H14: If developers have experience with similar applications, then prototyping will not be used.	0.10*	0.14
H15: If developers are not experienced with prototyping, then prototyping will not be used.	0.00***	0.15
H16: If a project does not have management support, then prototyping will not be used.	0.85	0.89
H17: If a need for experimentation and learning before full commitment exists, then prototyping will be used.	0.39	0.88
H18: If prototyping support tools are not available, then an evolutionary prototyping strategy will not be used.	0.02**	0.22

* $p \leq 0.10$ ** $p \leq 0.05$ *** $p \leq 0.01$	(+) relationship does not support hypothesis

4.8 Chapter IV Summary

This chapter has examined the data obtained from a mail survey designed to collect information to address the 19 pre-specified hypotheses.

First, the validity and reliability of the instrument used to collect the data was examined. It was demonstrated that the instrument is both valid and reliable.

Second, a profile of the 100 systems indicated that the sample is representative of the population from which they were drawn. Representativeness was tested by looking at the industry from which the respondents came from, as well as using the time of response and the demographics of company size, department size, and use of prototyping. All tests confirmed that the sample is representative.

Next, each of the hypotheses was individually investigated using various statistical techniques. Of the 18 hypotheses, five could not be proven false. Also, Hypothesis 19, the effect on system success, was also examined as part of each of the 18 hypotheses.

In the next chapter, the findings listed in this chapter will be interpreted (i.e., what do these results mean, and what are the implications?).

CHAPTER V

DISCUSSION

5.1 Overview

In this chapter, the results from the previous chapter will be interpreted and discussed. Also, supplemental analyses which were unnecessary for testing the hypotheses in the previous chapter, but needed to explain the results, are presented in this chapter.

5.2 Interpreting Hypothesis Tests

5.2.1 Selecting a Prototyping Strategy

In the previous chapter, it was discovered that only five of the possible 18 contingencies for selecting a prototyping strategy are actually followed in practice. The interpretation of those five contingencies are: (1) prototyping is used when developers have less experience with the application; (2) prototyping is used when developers have more experience with the use of prototypes; (3) user participation is higher when prototyping is used; (4) the use of prototyping is related to the user's prior knowledge of prototyping; and (5) prototyping tools are utilized when using an evolutionary prototyping strategy.

In Table 63, some measures of association between the factors and the suggested prototyping strategies for each of the five contingencies are presented.

TABLE 63

MEASURES OF ASSOCIATION FOR THE FIVE PROVEN CONTINGENCIES

Factor	Prototyping Strategy	r	R^2
User participation	prototyping	0.30	0.09
User PT knowledge	prototyping	0.29	0.08
Developer's experience w/ similar applications	prototyping	-0.17	0.03
Developer's PT knowledge	prototyping	0.65	0.42
Use of tools	Evolutionary	0.24	0.06

The correlations shown in Table 63 confirm the results found using t-tests and contingency tables in Chapter 4 (all correlations are significant at $p \leq 0.10$). Also illustrated in Table 63 are the variances explained by each factor in the choice of a prototyping strategy (i.e., R^2). With the exception of "Developer's PT knowledge," the individual factors have very little power in explaining the use of the respective prototyping strategies. Collectively, the first four factors explain 51 percent of the variance in choosing a "yes/no" prototyping strategy ($p \leq 0.0001$).

5.2.2 Influence on System Success

Hypothesis 19 suggests that the proper use of prototyping, as determined by each of the 18 contingencies, will have a positive effect on system success. Overall, as determined in this study, this hypothesis is not true. Of the five contingencies that are followed in practice, compliance with the contingency did not seem to affect system success. Of the remaining contingencies which were not followed in practice, only two would have affected system success if followed. Both are discussed below.

System success was higher for projects that followed the suggestion of Hypothesis 8, which states that an evolutionary prototyping strategy should not be used for projects with stringent performance requirements. Hypothesis 8 was not followed in practice, but, according to the influence on system success, should be followed.

System success was also higher for projects that followed the guideline: if there are a large number of users, then prototyping will not be used (Hypothesis 12). Unfortunately, the data shows that this guideline was not followed in practice. It can therefore be concluded that the use of prototyping when a large number of users are involved will result in a lower likelihood of system success.

5.3 Measures of Association

In Section 5.2, the association between the variables and the prototyping strategy was examined for the hypotheses that were previously determined to be significant. In this section, an analysis of the relationships among all of the variables is provided.

Correlation analysis was used to determine the association among the variables used in this study (i.e., from each of the hypotheses, prototyping strategy, and system success). The results are presented in Table 64. Overall, we could expect to find, by chance, 24 correlations significant at 0.10 (i.e., 242 correlations x 0.10 = 24.2 correlations by chance). As seen in Table 64, 47 correlations were found to be significant at $p \leq 0.10$. The probability of finding 47 or more significant correlations by chance is less than 0.00001.

5.3.1 Interpretations of Correlations

For brevity, only a few of the more "interesting" correlations shown in Table 64 will be discussed here.

System success is related to the user's level of participation in the development of the system (r=0.18). This is in agreement with many other studies (e.g., Edstrom, 1977; Ives and Olson, 1984; Tait and Vessey, 1988).

TABLE 64

CORRELATION MATRIX

Variable [Hypothesis]		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.
1. PT Strategy		-										
2. Success		.18	-									
3. Clarity of Requirements	[H1]	-.00	-.09	-								
4. Requirements Stability	[H2]	.04	-.13	.52***	-							
5. System Mode	[H3]	.04	.04	-.07	-.07	-						
6. Duration (months)	[H4]	.12	.11	.13	.30***	-.00	-					
7. Innovativeness	[H5]	-.16	-.01	-.02	.00	.03	-.04	-				
8. Percentage of Budget	[H6]	-.18	-.50***	.07	.10	-.15	.07	.42***	-			
9. Development man-hours	[H6]	.05	.11	.06	.19*	.09	.59***	-.02	.00	-		
10. System Rank (in cost)	[H6]	-.06	-.09	-.19*	-.16	-.25***	-.21**	.02	.04	-.18	-	
11. Critical/Non-critical	[H7]	.16*	-.13	-.04	-.05	-.11	-.12	-.02	-.25	-.11	.05	-
12. Performance Requirements	[H8]	-.11	-.19*	.08	.02	-.00	-.33***	.11	-.03	-.17	.36***	.25***
13. User Participation	[H9]	-.30***	-.18*	.11	.03	-.01	-.22**	-.10	.15	-.12	.07	.03
14. User Prototyping Knowledge	[H10]	.29***	.13	-.19*	-.01	-.04	.14	.03	.11	.23**	.05	.23**
15. User Automation Experience	[H11]	.14	.11	-.03	.05	-.03	.09	.04	.22	.14	-.01	.12
16. # of Users in Development	[H12]	.00	-.04	-.09	.07	-.10	.28***	.03	.83***	.16	-.04	-.07
17. Impact on User	[H13]	-.01	-.29***	.05	.28***	-.03	.33***	.04	.31*	.16	-.09	-.30***
18. Dev App Experience	[H14]	-.17*	.02	-.26***	-.10	.05	-.09	.28***	.21	-.16	-.08	.02
19. Dev PTing Experience	[H15]	.65***	-.04	-.03	.07	-.09	-.02	-.21**	-.21	.00	.12	.16
20. Management Support	[H16]	.02	.16	-.13	-.16	.05	-.00	.17*	.20	.02	.13	-.03
21. Need for Experimentation	[H17]	-.09	.07	-.19*	-.33***	.06	-.19*	-.03	-.09	-.14	.01	.02
22. Use of Prototyping Tools	[H18]	-.06	.09	.22*	.05	-.10	-.09	.00	.01	-.10	-.17	-.07

TABLE 64 (continued)

Variable [Hypothesis]		12.	13.	14.	15.	16.	17.	18.	19.	20.	21.	22.
13. User Participation	[H9]	.17*	-									
14. User Prototyping Knowledge	[H10]	-.11	-.27***	-								
15. User Automation Experience	[H11]	-.07	-.10	.42***	-							
16. # of Users in Development	[H12]	-.16	-.14	.36***	.21**	-						
17. Impact on User	[H13]	-.02	-.15	.00	-.03	.20*	-					
18. Dev App Experience	[H14]	-.12	-.06	-.01	.03	-.05	.02	-				
19. Dev PTing Experience	[H15]	-.07	-.12	.21**	-.11	-.02	-.05	-.06	-			
20. Management Support	[H16]	-.06	-.13	.20**	.13	.12	-.14	.04	-.09	-		
21. Need for Experimentation	[H17]	.10	-.09	-.24**	-.09	-.05	-.06	.08	-.08	.03	-	
22. Use of Prototyping Tools	[H18]	-.14	-.02	-.03	.09	.09	-.07	-.14	.05	.04	-.11	-

* $p \leq 0.10$
** $p \leq 0.05$
*** $p \leq 0.01$

Although the correlation shown is negative, the relationship is positive (the negative correlation is a result of the scaling direction of the question).

System success is also associated (negatively) with the impact on the user. In this case, lower levels of impact (i.e., less reorganization, less job changes) are related to higher levels of success (i.e., satisfaction), and vice versa. This would imply that the user is more satisfied with systems that require fewer changes in their environment.

Most of the other correlations were expected and self-explanatory, and do not lend insight into the discussion of results. For example, project duration (in months) is related to the man-hours needed to develop the system, system rank (in cost), performance requirements, user participation, number of users participating in development, and impact on the user. This implies (and makes sense) that larger systems require higher performance requirements, require more users and user participation in development, and impact the user, more than smaller systems.

5.4 Prototyping and System Success

Success, as defined in this study, is determined by the user's satisfaction. It is suspected, based upon the results of previous studies (e.g., Edstrom, 1977; Ives and Olson, 1984; Tait and Vessey, 1988), that the satisfaction would be higher for systems developed using a prototyping approach, because of the increased user involvement. The findings from this study indicate that, although the satisfaction is higher, the difference is not significant (see Table 65). However, it must be noted that the assumption (for t-tests) of equal variances was violated, therefore the Cochran statistic was used.

TABLE 65

PROTOTYPING'S RELATIONSHIP TO UIS

Prototyping?	N	Mean	Prob>|T|
NO	27	5.64	0.12
YES	68	7.54	

Budget and development time are also considered to be measures of success (DeLone and McLean, 1992). A system is considered a failure if it is over-budget or delivered late (Lyytinen, 1987); although these measures are not as common as UIS (DeLone and McLean, 1992).

Respondents were asked to provide information regarding the original budget of the system and the final cost of the completed system. To determine the budget status (i.e., over- or under-budget), the original budget was subtracted from the final cost. A positive number indicates over-budget; a negative number indicates under-budget. A t-test was performed using the calculated budget status between projects using prototyping and projects not using prototyping. The results are shown in Table 66.

TABLE 66

PROTOTYPING'S RELATIONSHIP TO BUDGET STATUS

Prototyping?	N	Mean*	Prob>|T|
NO	8	$15,454	0.16
YES	73	$79,041	

* positive mean indicates over-budget
  negative mean indicates under-budget

Regardless of the use of prototyping, the projects were (on average) over-budget. Projects that did not use prototyping were over-budget an average of $15,454; projects that used prototyping were over-budget an average of $79,041. However, due to the low response rate to these questions, and the large variance of the means, the difference is not significant. With a small sample size, it is likely that a difference existed, but was not detectable.

Respondents were also asked to provide the number of months needed to develop the system, and the number of months originally anticipated. The difference between the anticipated months and the actual months indicates the delivery status of the system (i.e., late or not late). A t-test, as shown in Table 67, indicates no significant difference between the delivery status of projects using prototyping, and those not using prototyping. Generally, all projects reported in this study were delivered late. However, the use of prototyping did not seem to have an effect on the delivery status of a project.

TABLE 67

PROTOTYPING'S RELATIONSHIP TO DELIVERY STATUS

Prototyping?	N	Mean	Prob>|T|
NO	21	4.38	0.99
YES	63	4.35	

* positive mean indicates late delivery
negative mean indicates early delivery

5.5 An Investigation of the Conceptual Model

In Chapter 2, a conceptual model was derived from the literature as a means of explaining the relationships investigated in this study. For convenience, the model is presented again in Figure 2.

The model in Figure 2 explicitly suggests two relationships: the impact of various characteristics on the choice of a prototyping strategy, and the influence of a properly chosen prototyping strategy on system success. To this point in the study, these two relationships have been examined. As previously demonstrated, five of the suggested 18 factors have an effect on the selection of a prototyping strategy. The second relationship was examined as Hypothesis 19. However, the model also implies a third relationship: the effect of the various characteristics on system success as moderated by prototyping strategy. A moderator variable is defined as one which modifies either the form and/or strength of the relationship between a predictor and a criterion variable (Sharma, Durand, and Gur-arie, 1981). In this study, the predictor variables are the various characteristics, and the criterion variable is system success.

Moderated hierarchical regression was performed to assess the moderating effect of prototyping strategy on the various characteristics to system success. First, system success was regressed on the combination of a single factor and prototyping strategy. The next step added the interaction term of the factor and prototyping strategy. The incremental contributions of R^2 provided by the interaction term added to the model was then evaluated. If

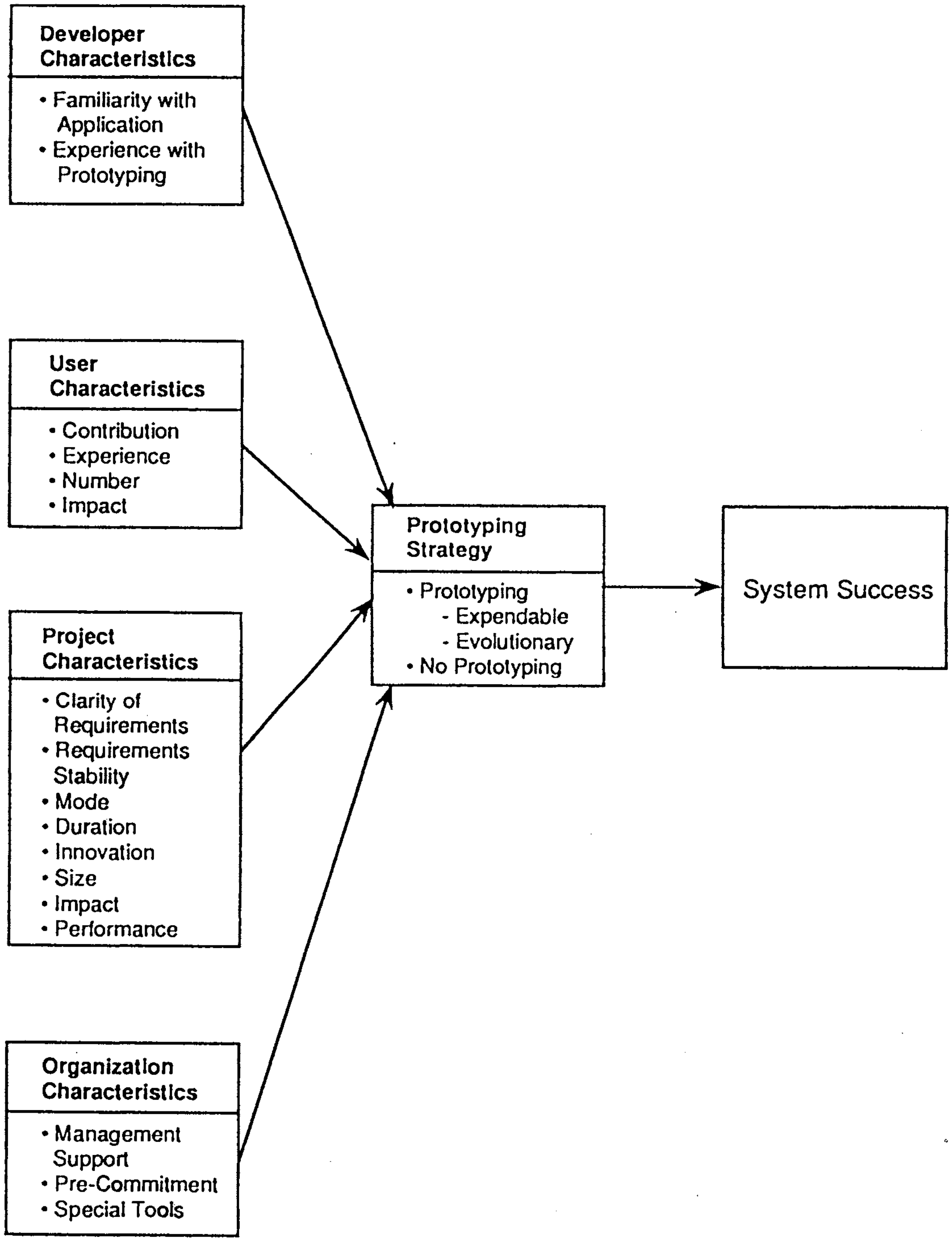

Figure 2. Conceptual Model of Prototyping Strategy Selection

the incremental change in R^2 is significant on the steps when the interaction terms were entered, then the interactions accounted for a significant portion of the total explained variance beyond the main effects (i.e., it can be concluded that the relationship between the variable and system success is moderated by the prototyping strategy) (Eastman, 1990). The results of the moderated hierarchical regression analysis are provided in Table 68.

TABLE 68

MODERATED HIERARCHICAL REGRESSION RESULTS:
CHARACTERISTICS ON SYSTEM SUCCESS

Characteristic	R^2	ΔR^2
Clarity of Requirements + PT Strategy	0.05	
+ Clarity of Requirements*PT Strategy	0.05	0.00
Requirements Stability + PT Strategy	0.05	
+ Requirements Stability*PT Strategy	0.06	0.01
System Mode + PT Strategy	0.03	
+ System Mode*PT Strategy	0.04	0.01
Development Time + PT Strategy	0.06	
+ Development Time*PT Strategy	0.08	0.02
Innovativeness + PT Strategy	0.04	
+ Innovativeness*PT Strategy	0.06	0.02
Percent-of-budget + PT Strategy	0.25	
+ Percent-of-budget*PT Strategy	0.34	0.09*
Man-hours + PT Strategy	0.05	
+ Man-hours*PT Strategy	0.06	0.01
Rank (in cost) + PT Strategy	0.06	
+ Rank*PT Strategy	0.06	0.00

TABLE 68 (continued)

Characteristic	R^2	ΔR^2
Criticalness + PT Strategy + Criticalness*PT Strategy	0.07 0.12	 0.05***
User Participation + PT Strategy + User Participation*PT Strategy	0.05 0.10	 0.05**
User PT Knowledge + PT Strategy + User PT Knowledge*PT Strategy	0.04 0.07	 0.03*
Number of Users + PT Strategy + Number of Users*PT Strategy	0.04 0.07	 0.03*
User Automation Experience + PT Strategy + User Automation Experience*PT Strategy	0.04 0.05	 0.01
User Impact + PT Strategy + User Impact*PT Strategy	0.12 0.12	 0.00
Developer's App Exp + PT Strategy + Developer's App Exp*PT Strategy	0.04 0.04	 0.00
Developer's PT Knowledge + PT Strategy + Developer's PT Knowledge*PT Strategy	0.10 0.12	 0.02
Management Support + PT Strategy + Management Support*PT Strategy	0.06 0.08	 0.02
Experimentation + PT Strategy + Experimentation*PT Strategy	0.05 0.05	 0.00

* $p \leq 0.10$
** $p \leq 0.05$
*** $p \leq 0.01$

As shown in Table 68, five interactions are significant: (1) percent-of-budget and prototyping strategy; (2) criticalness and prototyping strategy; (3) user participation and prototyping strategy; (4) user's knowledge of prototyping and prototyping strategy; and (5) number of users during development and prototyping strategy. The interactions are graphed and presented in Figures 3 through 7.

To test Hypothesis 19 during the previous chapter, the relationship between each of the 18 individual factors and the prototyping strategy was investigated for their combined effect on system success. Thus, in a "brute force" way, the interactions were examined. But, the results of Hypothesis 19 only suggested a relationship between a factor and the prototyping strategy and system success - direction was not examined. The interactions presented in this section go beyond Hypothesis 19 and investigate the specific effects of the various factors and prototyping strategy on system success. Therefore, we would expect to find some similarities between the results of Hypothesis 19 and the results presented in the this section. For each of the interactions discussed below, the relationship to Hypothesis 19 will be evaluated.

5.5.1 Percent-of-Budget and Prototyping Strategy

The graph in Figure 3 illustrates the interaction of percent-of-budget and prototyping strategy. Visually, the interaction suggests that success is higher when prototyping is used to develop small (i.e., lower percent-of-budget) systems, and vice versa. However, because a visual inspection of an interaction can sometimes be misleading, an examination of the simple effects is used to provide a better explanation of the interaction.

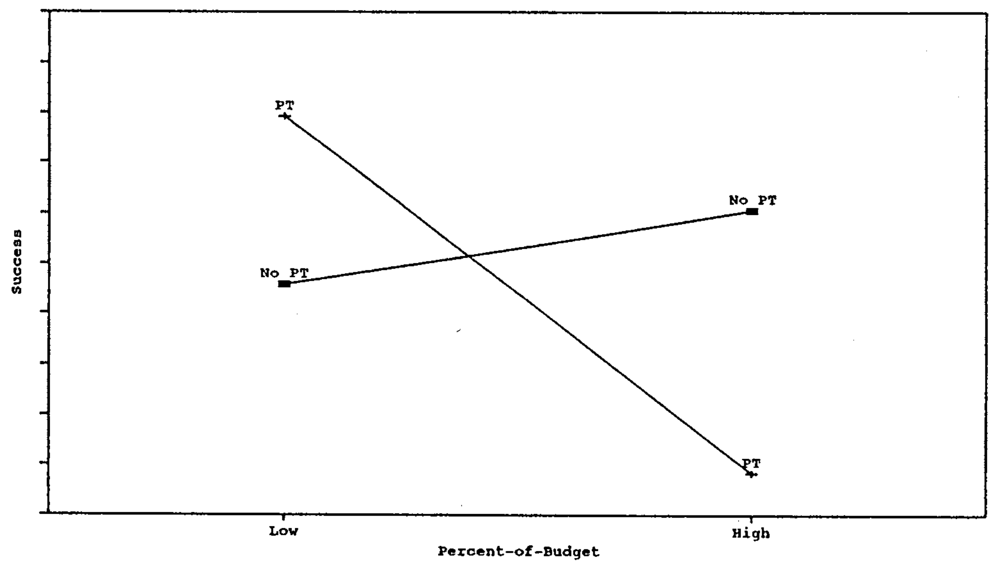

Figure 3. Interactions of Percent-of-Budget and Prototyping Strategy

For the interaction presented in Figure 3, a test of the simple effects for a two-way interaction provides significance levels for changes in the following: (1) the slope of the "PT" line, which indicates the effect on the dependent variable (SUCCESS) when prototyping is used on the range of low percent-of-budget to high percent-of-budget; (2) the slope of the "No PT" line, which indicates the effect on the dependent variable (SUCCESS) when no prototyping is used in the range of low percent-of-budget to high percent-of-budget; (3) the difference in SUCCESS between "PT" and "No PT" for a LOW percent-of-budget; and (4) the difference in SUCCESS between "PT" and "No PT" for a HIGH percent-of-budget. The simple effects for the percent-of-budget and prototyping strategy interaction are provided in Table 69.

TABLE 69

SIMPLE EFFECTS OF PERCENT-OF-BUDGET AND PROTOTYPING STRATEGY INTERACTION

Effects Comparison	p-value
PT and percent-of-budget	0.0010
NO PT and percent-of-budget	0.6638
LOW percent-of-budget and PT strategy	0.0634
HIGH percent-of-budget and PT strategy	0.0012

The results of the simple effects test indicate that: (1) success is significantly higher when prototyping is used for low percent-of-budget projects compared to high percent-of-budget projects (p = 0.0010); (2) there is no difference in success between high and low percent-of-budget projects when prototyping is not used (p = 0.6638); (3) for low percent-of-budget projects, success is significantly higher when prototyping is used (p = 0.0634); and (4) for high percent-of-budget projects, success is significantly higher when prototyping is not used (p = 0.0012). Thus, these findings are contrary to the suggested contingency of using prototyping for larger systems (Hypothesis 6). However, the simple effects test reinforces findings presented earlier in Section 4.7.6 (Hypothesis 19) which indicated that success is higher when prototyping is used for smaller percent-of-budget projects. In Section 4.7.6, the small sample size prevented a definite conclusion.

In Chapter 2, it was demonstrated that "system size" was a highly debated contingency among researchers - some advocate using prototyping only for small systems, others would suggest using prototyping for large systems. The findings presented in this section suggest that prototyping should be used for small systems, and not used for large systems. From the literature, reasons supporting the findings here include: (1) for large systems, designers lack the detailed documentation that other methods provide (Yaverbaum, 1989); (2) for large systems, it becomes difficult to prototype the entire project, and managing the system development process becomes difficult (Burns and

Dennis, 1985; Carey, 1990; Dennis, Burns, and Gallupe, 1987; Klinger, 1986, 1988; Lynch, 1987; Mahmood, 1987).

Based upon these results and the supporting literature, it would seem that prototyping is not appropriate for large systems. Prototyping, as a methodology, lacks the proper supporting documentation and management discipline necessary to produce a successful system. For smaller systems, these aspects are not as important, thus, prototyping can be used effectively.

5.5.2 Criticalness and Prototyping Strategy

Figure 4 demonstrates the interaction of prototyping strategy and criticalness (i.e., critical/non-critical). The results of the simple effects test are presented in Table 70. The information provided in Figure 4 and Table 70 indicate the following: (1) success is not affected by the use of prototyping between critical and non-critical systems ($p = 0.7113$); (2) success is significantly higher when prototyping is not used for critical systems compared to non-critical systems ($p = 0.0018$); (3) for critical systems, success is not affected by the prototyping strategy chosen ($p = 0.5216$); and (4) for non-critical systems, success is significantly higher when prototyping is used ($p = 0.0043$). Similar conclusions were reached earlier in Hypothesis 7 when it was shown that, in practice, prototyping was used for the development of non-critical systems, and that subsequent system success was higher (although not significantly higher).

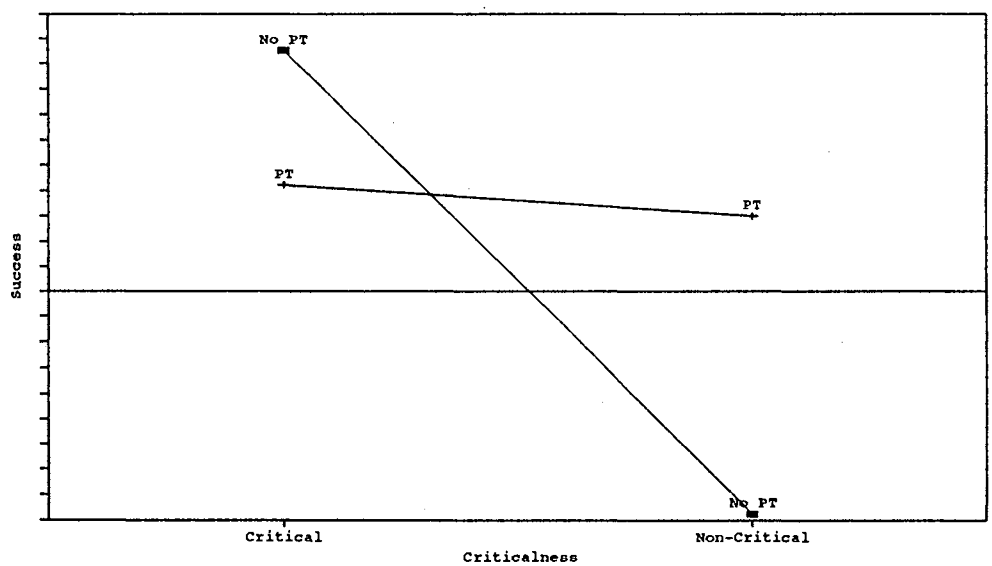

Figure 4. Interactions of Criticalness and Prototyping Strategy

TABLE 70

SIMPLE EFFECTS OF CRITICALNESS AND PROTOTYPING STRATEGY INTERACTION

Effects Comparison	p-value
PT and criticalness	0.7113
NO PT and criticalness	0.0018
Critical and PT strategy	0.5216
Non-critical and PT strategy	0.0043

Discussions in the literature concerning the use of prototyping and criticalness unanimously concur that prototyping should be used for critical systems (Boar, 1986; Carey, 1990; Dos Santos, 1988; Gremillion and Pyburn, 1983). These researchers contend that for critical systems it is extremely important that the system meet specifications and prototyping facilitates this important requirement. However, none of the aforementioned studies are empirically based. Thus, although intuitively it may seem that prototyping should be used for critical systems, in reality, the data does not suggest that prototyping is more successful than not using prototyping. In fact, the interaction suggests that for critical systems, the prototyping strategy does not affect success. For non-critical systems, prototyping produces a more successful system than non-prototyped systems.

Why would prototyping a critical system not produce a more successful system than a non-prototyped critical system? We can postulate that due to the nature of a critical system (i.e., operates, manages, and controls the daily business activities), the development team will take all measures possible to ensure that the specifications of the system are met. The system is simply too important to the organization to be developed poorly. Therefore, regardless of the prototyping strategy, the development team will do their best to meet specifications. Why, then, would the success of a non-critical systems be affected by the prototyping strategy? Perhaps, for non-critical, less attention is paid to meeting specifications by the developers, thus, the use of prototyping is better able to assist the developers in meeting specifications.

5.5.3 User Participation and Prototyping Strategy

Figure 5 shows the interaction between prototyping strategy and user participation on system success. Table 71 provides the results of the simple effects test. Based on the information provided by the interaction, the following conclusions can be drawn: (1) success is not affected by the use of prototyping for any level of user participation ($p = 0.8264$); (2) success is significantly higher when prototyping is not used for high levels of user participation compared to low levels of user participation ($p = 0.0111$); (3) for low levels of user participation, success is significantly higher when prototyping is used ($p = 0.0272$); and (4) for high levels of user participation, success is not affected by the prototyping strategy chosen ($p = 0.7635$). This relationship contradicts Hypothesis 9 which suggests that prototyping should not be used when user participation is low (See Section 4.7.9). However, the interaction supports the results presented in Section 4.7.9 (Hypothesis 19), which found that success was affected by the level of user participation depending on the prototyping strategy (although the relationship was not significant). See Table 43 for more information.

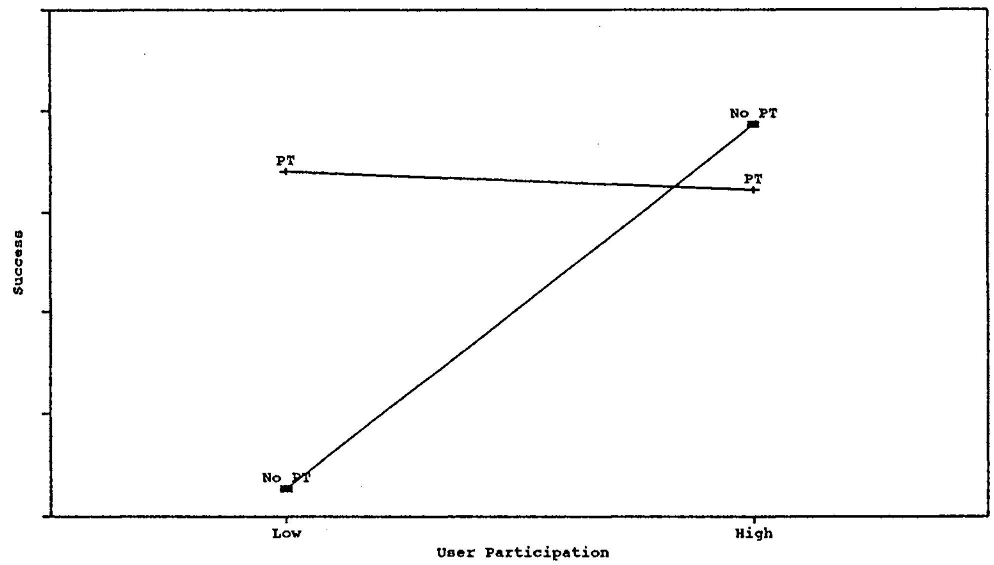

Figure 5. Interactions of User Participation and Prototyping Strategy

TABLE 71

SIMPLE EFFECTS OF USER PARTICIPATION AND PROTOTYPING STRATEGY INTERACTION

Effects Comparison	p-value
PT and user participation	0.8264
NO PT and user participation	0.0111
Low User Participation and PT strategy	0.0272
High User Participation and PT strategy	0.7635

Many studies have demonstrated that increased user involvement improves the user's satisfaction with the system (e.g., Franz and Robey, 1986; Tait and Vessey, 1989). Prototyping, compared to the traditional SDLC, requires more communication and interaction between the user and developer. Thus, it has been postulated that prototyping facilitates higher user participation (in fact, requires high user participation), and should be used with high user participation to increase the likelihood of system success (Gibson and Rademacher, 1987; Meyer and Kovacs, 1983; Teng and Sethi, 1990). The findings presented in this section do not confirm those postulations. It is demonstrated here that at high levels of user participation, the success of the system is not affected by the prototyping strategy. However, at low levels of participation, the use of prototyping produces a more successful system compared to not using prototyping. Possible reasons for these findings follow.

Prototyping is a means of obtaining high user participation. However, if the participation can be obtained without using prototyping, system success can still be achieved. Therefore, prototyping strategy does not affect system success for high levels of user participation. The important factor seems to be obtaining a high level of user participation.

For low levels of user participation, prototyping provides a mechanism for easy communication between developers and users. Whereas the traditional SDLC requires users to read and understand complicated analysis and design documents, prototyping only requires the users to view a prototype of the system and verify its usefulness (or correctness). Thus, as seen here, prototyping is more successful than non-prototyping for low levels of user participation.

5.5.4 User Knowledge of PT and Prototyping Strategy

Figure 6 shows the interaction of a user's knowledge of prototyping and prototyping strategy. Table 72 presents the results of the corresponding simple effects test. Figure 6 and Table 72 suggest the following relationships: (1) success is not affected by the use of prototyping for any level of a user's prototyping knowledge ($p = 0.4055$); (2) success is significantly higher for high levels of a user's prototyping knowledge compared to low levels of knowledge when prototyping is not used ($p = 0.0839$); (3) for low levels of a user's knowledge of prototyping, success is significantly higher when prototyping is used ($p = 0.0064$); and (4) for high levels of a user's prototyping knowledge, success is not affected by the prototyping strategy chosen ($p = 0.8256$).

These results contradict Hypothesis 10 which suggests that prototyping should not be used when users have a low level of prototyping knowledge. However, the interaction supports the results presented in Section 4.7.10 which found that success is affected by the use of prototyping when the user's knowledge is considered. The interaction provides a more specific conclusion: success is higher when prototyping is used with users of low prototyping knowledge; for high levels of knowledge, prototyping strategy does not affect system success.

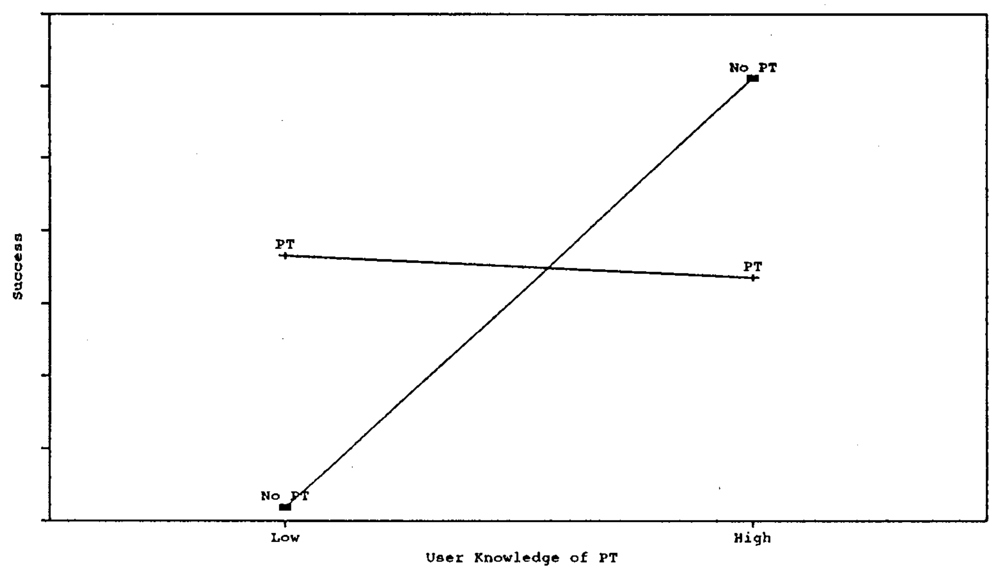

Figure 6. Interactions of User Knowledge of PT and Prototyping Strategy

TABLE 72

SIMPLE EFFECTS OF USER KNOWLEDGE OF PT AND PROTOTYPING STRATEGY INTERACTION

Effects Comparison	p-value
PT and user knowledge of PT	0.4055
NO PT and user knowledge of PT	0.0839
Low User Knowledge of PT and PT strategy	0.0064
High User Knowledge of PT and PT strategy	0.8256

"Users who are not familiar with prototyping often have unrealistic expectations" is a common argument by researchers who advocate that users should have prior knowledge of prototyping before utilizing a prototyping approach (Alavi, 1984a; Berrisford and Wetherbe, 1979). However, as presented in this section, system success is not affected by the prototyping strategy at a high level of familiarity with prototyping. Intuitively, this result makes sense. That is, a high knowledge level of prototyping shouldn't affect the outcome of a project, regardless of the prototyping strategy. Consider the following analogy: Joe is very knowledgeable of vehicular transportation (i.e., driving a car). If Joe needs to take a trip, then driving a car would provide a satisfactory form of transportation and resulting trip (i.e., using prototyping when the user has a knowledge of prototyping would provide a successful system). However, if Joe decides to fly to his destination, he can still be satisfied with the mode of transportation and the resulting trip (i.e., even though a user has a knowledge of prototyping, an alternative strategy can be used and still produce a successful system).

The use of prototyping is discouraged when the users do not have knowledge of the prototyping approach. As previously mentioned, the reason is unrealistic expectations. For example, anything shown to the user in the form of a prototype may be perceived to be a fully operational system. However, as evidenced in this section, for a low level of prototyping knowledge, system success is higher when prototyping is used, compared to not using prototyping. Thus, it would appear, as long as the prototyping process is managed properly, prototyping can be successfully used when users do not have knowledge of the process. Even though users may not understand prototyping, they are excited to see something (i.e., a prototype). If the developers can "manage" the users expectations, the use of prototyping is the best alternative.

5.5.5 Number of Users and Prototyping Strategy

Figure 7 shows the interaction between the number of users and prototyping strategy, and Table 73 provides the simple effects tests. The data provided in Figure 7 and Table 73 suggest: (1) success is significantly higher when prototyping is used for a small number of users compared to using prototyping for a large number of users (p = 0.0212); (2) success is marginally higher when prototyping is not used for a large number of users compared to a small number of users (p = 0.1048); (3) for a small number of users, success is not affected by the prototyping strategy (p = 0.9871); and (4) for a large number of users, success is significantly higher when prototyping is not used (p = 0.0779).

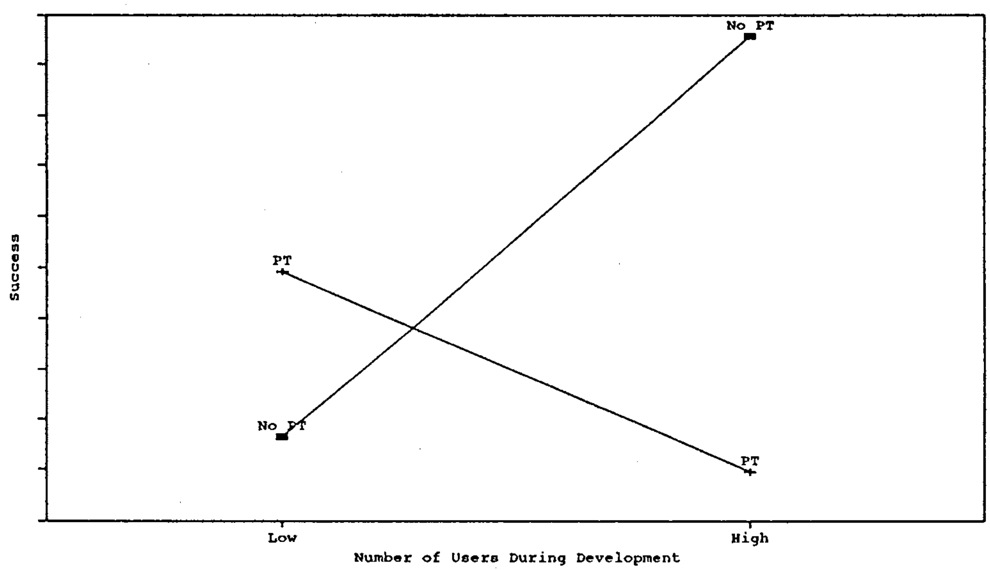

Figure 7. Interactions of Number of Users and Prototyping Strategy

These results reinforce the contingency suggested by Hypothesis 12, and the findings presented in Section 4.7.12. In Table 49 presented earlier, it was demonstrated that system success was lower for projects that involved a large number of users in development when prototyping was used.

TABLE 73

SIMPLE EFFECTS OF NUMBER OF USERS AND PROTOTYPING STRATEGY INTERACTION

Effects Comparison	p-value
PT and Number of Users	0.0212
NO PT and Number of Users	0.1048
Low Number of Users and PT strategy	0.9871
High Number of Users and PT strategy	0.0779

Involving a large number of users in the evaluation and alteration of a prototype, and in generating requirements, would be difficult to coordinate. Prototyping for one or a few users is not difficult; prototyping for many is (Burns and Dennis, 1985). Modifications requested by one user must be approved by all affected users; iterations become slower and change is no longer a quick and easy task. Managing changes, and requests for changes, becomes almost impossible. Tillman (1989) refers to this as the "internal consistency nightmare."

The delays in development and slow response times irritate the users, which results in lower user information satisfaction (i.e., system success). Gavurin (1991) attempted to use small random samples of users at each iteration as a way to control the number of users, but found that it was not an effective method because new users at each iteration would provide requirements that conflicted with users of earlier iterations. If the new requirements were implemented, the first set of users were dissatisfied. If the new requirements were not implemented, the second set of users were dissatisfied - the process becomes a "no win" situation. Therefore, as the results in this section indicate, it is best to use a "no prototyping" strategy if a large number of users are involved.

5.5.6 Validity of the Conceptual Model

Earlier, it was established that only five factors are moderated by prototyping. In addition to the five significant interactions, three main effects were significant: (1) prototyping strategy; (2) impact; and (3) developer's prototyping knowledge. Therefore, as a means of validating the conceptual model, a regression model was built using the five interactions and three main effects. The results are shown in Table 74.

Interestingly, the main effects and the interactions that were significant on an individual basis were not significant collectively. The model significance level is 0.2749, and none of the variables are individually significant. Obviously, this does not support the existence of the conceptual model.

TABLE 74

REGRESSION OF CONCEPTUAL MODEL
VARIABLES ON SYSTEM SUCCESS

Variable	Parameter Estimate	Prob > [T]
Intercept	5.13	0.27
PT Strategy	3.11	0.45
Impact	-0.30	0.68
Developer's PT Knowledge	0.72	0.47
Percent-of-Budget*PT Strategy	-1.84	0.49
Criticalness*PT Strategy	-1.17	0.53
User Participation*PT Strategy	-0.26	0.72
User Knowledge of PT*PT Strategy	0.03	0.96
Number of Users*PT Strategy	-0.08	0.44

Prob > F: 0.2318
R-square = 0.35

Why would the five significant interactions and three significant main effects not provide an explanatory model that, overall, is significant? The primary reason is multicollinearity. Multicollinearity arises from patterns of strong intercorrelations among the independent variables. In other words, individually, the eight independent variables (five interactions and three main effects) do a good job of explaining success, collectively, however, the explaining power of the variables overlap so much that the overlap results in very large standard errors of the estimated coefficients, suggesting statistical insignificance even when the relationships involved are quite strong. Does this imply that we must view the characteristics individually in order to explain success? Certainly not! This only suggests that ALL of the characteristics of the project should be considered and that a project does not exist with only one or a few of the characteristics in isolation. Concerning the validity of the conceptual model, this indicates that only a few variables are moderated by prototyping. Thus, overall, the conceptual model is invalid.

As determined, the variables should not be considered in isolation. Therefore, as a final attempt at examining the conceptual model, a stepwise regression technique was built by including all variables from the study and all interactions between each variable and prototyping strategy. Variables were allowed to enter the model at a generous significance level of 0.15. The results are shown in Table 75.

Eight variables entered at a significance level of 0.15. No interactions entered the model, and, surprisingly, prototyping strategy did not enter the model. Overall, the model significantly accounts for the dependent variable's behavior (Prob > F = 0.0105). Sixty-two percent of the variance in system success is explained by these eight variables.

TABLE 75

STEPWISE REGRESSION: ALL VARIABLES AND INTERACTIONS ON SYSTEM SUCCESS

Variable	Parameter Estimate	Prob > [T]
Intercept	12.18	0.00
Clarity of Requirements	0.91	0.03
System Mode	-1.53	0.01
Innovativeness	-0.89	0.05
Man-hours	0.00	0.00
User Participation	-0.98	0.01
User's PT Knowledge	-1.74	0.00
Developer Application Experience	0.87	0.01
Need for Experimentation	1.90	0.12

Prob > F: 0.0105
R-square: 0.62

These eight variables do a good job of explaining the variance in system success. However, these variables were selected for inclusion in this model by their ability to explain success without greatly overlapping other variables.

In other words, these eight variables have the least in common with each other (low multicollinearity), yet the most power in explaining system success. Obviously, these variables are a result of this study only and should not be generalized for all applications.

Unfortunately, the results presented in the Tables 74 and 75 do not validate the conceptual model used in this study. Instead, the results suggest that the model is not valid. It would seem that: (1) a variety of characteristics affect system success; (2) prototyping strategy does not moderate the relationship between all of the variables and system success; and (3) prototyping strategy does not individually have an effect on system success.

5.6 Development of a Contingency Model

This study started with two main purposes: (1) gather evidence which will indicate the characteristics influencing the selection of a prototyping strategy; and (2) propose a contingency model of prototyping strategy selection based upon the results of Purpose 1. The first objective has been achieved and demonstrated earlier in this chapter and in Chapter 4. The second objective, proposing a contingency model, is discussed next.

Contingency models identify alternatives based upon situations. In this case, the alternatives are the prototyping strategies - prototyping (expendable, evolutionary), and no prototyping. The situations are the characteristics surrounding the development effort. For this study, two contingency models are constructed. One model is based upon Hypotheses 1 through 18 which show how prototyping is used in practice in industry. The second model is based on the proper selection of a prototyping strategy and its influence on system success. The two contingency models are provided in Tables 76 and 77, respectively.

The contingency model in Table 76 illustrates the characteristics which influence the selection of a prototyping strategy. Table 76 is based on the data presented in Hypotheses 1 through 18 which reflect the way prototyping is used in practice. In summary, Table 76 shows that prototyping (either expendable or evolutionary) involves a high level of user participation; and is used: (1) when users have a high level of prototyping knowledge; (2) when developers have a high level of prototyping knowledge; (3) when developers do not have experience with similar applications; and (4) for non-critical systems. Table 76 also shows that evolutionary prototyping requires special prototyping tools.

Once again, the contingencies in Table 76 are based upon the evidence illustrated for Hypotheses 1 through 18 which demonstrate the use of prototyping in industry. Admittedly, this contingency model probably has very little value other than providing an organization with a view to how the industry in general is using prototyping.

TABLE 76

CONTINGENCY MODEL FOR SELECTING A PROTOTYPING STRATEGY, BASED ON INDUSTRY PRACTICE

	Prototyping Strategy		
Characteristic	Exp	Evol	None
User Participation	High	High	Low
User's PT Knowledge	High	High	Low
Developer Experience w/ Similar Applications	Low	Low	High
Developer Experience w/ PT	High	High	Low
Need for PT Tools	No	Yes	No
Critical Systems	No	No	Yes

The second contingency model, Table 77, is built from the selection of a prototyping strategy's influence on system success. This table was constructed from information provided earlier for the significant interactions between various factors and prototyping strategy (see Section 5.5). In summary, Table 77 shows that: (1) prototyping should be used for projects of small size and should not be used for projects of large size (based on percent-of-budget); (2) prototyping should be used for non-critical systems; (3) prototyping should be used when user participation is low; (4) prototyping should be used when users have a low knowledge of prototyping; and (5) prototyping should not be used when a large number of users are involved in the development process.

TABLE 77

CONTINGENCY MODEL FOR SELECTING A PROTOTYPING STRATEGY, BASED ON INFLUENCE ON SYSTEM SUCCESS

	Prototyping Strategy		
Characteristic	Exp	Evol	None
Percent-of-Budget			
Low	X	X	
High			X
Critical Systems			
Critical	X	X	X
Non-critical	X	X	
User Participation			
Low	X	X	
High	X	X	X
User Knowledge of PT			
Low	X	X	
High	X	X	X
Number of Users in Development			
Low	X	X	X
High			X

(X = proper strategy)

The results of the moderated hierarchical regression analysis must be considered when viewing the contingencies presented in Table 77. That is, the moderated hierarchical regression analysis demonstrated the independent nature of the characteristics and related prototyping strategy. Therefore, it must be noted that these are not the only factors that could (or should) be used in selecting a prototyping strategy. Instead, these are the factors that, independently, are moderated by prototyping. However, the value of the contingency table should not be discounted because of the independent nature of the factors. If used properly, Table 77 can be very useful in suggesting guidelines for the use of prototyping such that system success is maximized.

5.7 Chapter V Summary

In this chapter, the results of the hypothesis tests from the previous chapter were interpreted and expanded. First, the significant findings were further discussed and explained. Next, measures of association between all variables used in the study was provided. A look at the use of prototyping strategy and system success was provided next. In addition to the UIS used as a surrogate for system success in this study, budget and time schedule were also evaluated as being indicators of system success. Next, the conceptual model which was used to guide the study was examined by using moderated hierarchical regression. It was found that the conceptual model is not a sound model. Finally, two contingency models were developed. One of the models was based on the use of prototyping in practice; the other model was based on the prototyping strategy's influence on system success.

CHAPTER VI

CONCLUSIONS, IMPLICATIONS, AND SUGGESTIONS FOR FURTHER RESEARCH

6.1 Overview

This, the last chapter, is used to summarize the findings of this study, illustrate implications of the research, determine shortcomings of the study, and provide suggestions for further research.

6.2 Summary of Findings

The two major purposes of this study were: (1) gather evidence which will indicate the characteristics influencing the selection of a prototyping strategy (i.e., contingency relationships); and (2) propose a contingency model from the results. In total, 18 contingency relationships were identified from the literature - most of which were not empirically founded. This study has attempted to determine the extent to which these contingencies are followed in practice, and their influence on system success.

Only five of the possible 18 contingencies are followed in practice: (1) prototyping is used when developers have less experience with the application; (2) prototyping is used when developers have more experience with the use of prototypes; (3) user participation is higher when prototyping is used; (4) the use of prototyping is related to the user's prior knowledge of prototyping; and (5) prototyping tools are utilized when using an evolutionary prototyping strategy. However, the five contingencies had no effect on system success. It was subsequently determined that two of the original 18 contingencies, if followed, would have improved system success: (1) an evolutionary prototyping strategy should be used when a system has strict performance requirements; and (2) prototyping should not be used if there are a large number of users involved in the development. Overall, most of the contingencies are not followed in practice, and, even if they were, system success would not be affected.

Two contingency models were proposed from the results. One model used the contingencies followed in practice as a basis (Table 76). The second model used system success as a basis (Table 77). Practitioners can evaluate the models in the context of its basis. First, practitioners can compare their selection of a prototyping strategy against those suggested by the industry as a whole (Table 76). Second, if system success is the primary concern, the contingency model can assist in the selection of a prototyping strategy which will increase the likelihood of system success (Table 77).

Overall, the results of this study suggest that prototyping is just one of many factors that affect system success, and the selection of a prototyping strategy is not dependent upon many other characteristics. The contingencies that are suggested are not followed in practice and seem to have little impact on system success.

6.3 Implications

The results of this study impact two different communities: academicians and practitioners. For the academician doing research in the area of prototyping, this study provides insight into the many assertions made in the literature. In fact, it disputes many of the assertions. As a researcher, this should allow for a focusing of the research area. It has been unfortunate that so much of the research in prototyping has been led by suppositions and suggestions that have never been tested.

This study should "clear up" many of the misunderstandings of the use of prototyping. As demonstrated in Chapter 2 (the literature review), many contingencies have been provided and many benefits of prototyping have been suggested. This study has empirically investigated the contingencies and the purported benefits and has found almost all of the contingencies and claims to be unfounded.

This research should also prove useful to the practitioner. For organizations not using prototyping (approximately 27 percent), this study provides a set of guidelines based upon how other practitioners are using prototyping, and, perhaps, how prototyping should be used to improve system success. For organizations already using prototyping (approximately 73 percent), it provides a mechanism to gauge their use of prototyping against other organizations. It also provides the opportunity to personally assess how effectively they have been using prototyping by examining those contingencies that affect system success.

6.4 Limitations of the Study

The analysis and findings in this study are subject to several limitations. In this section, some of the principal sources of error will be examined and their implications assessed regarding the validity of the research design.

A major threat to internal validity would be the selection of the user and project leader by the MIS manager. It would be difficult to ensure that the appropriate users and project leaders are selected. Reliance on the MIS manager could bias the results (Ives and Olson, 1984).

Another threat to internal validity is that the hypotheses are tested with self-report data (Field, 1979; Vroom and Jago, 1978). This can result in two weaknesses (Edstrom, 1977): (1) reliance on people's memories for data; and (2) reliance on perceptions of people instead of direct observation. However, this threat has been somewhat reduced. First, most of the systems were implemented in the past two years, which should lessen the problem of poor memories. Second, most of the responses required factual, rather than perceptual, information.

This type of study could also introduce a history effect, since events occurring between system implementation and the time of the study could change the way the system is viewed by users or developers. This is especially true of the instrument used to measure system success - user information satisfaction.

Ideally, the user, project leader, and MIS manager would not discuss the questionnaire with each other. Communication among the subjects could result in a threat to construct validity - hypotheses guessing. In this case, subjects may try to guess the hypotheses being tested and supply supporting data.

Lastly, a source of error could result from the user partially providing information for both the independent and dependent variables. In this case, the user is presenting information regarding user characteristics and satisfaction. Because user characteristics are independent variables and satisfaction is a dependent variable (a measure of system success), bias could be introduced.

The sample size is also a limitation in this study. The low sample size hampers the generalizability of the study. However, since this is a study of relationships among variables and not a description of any given population, satisfactory interpretation can be made.

Additionally, the sample has proven to be representative of the population.

Another limitation is the measure of system success. Although an undisputed definition and measure of success does not exist, the most popular measure was used in this study. However, the reluctance of organizations to report "unsuccessful" systems has resulted in a skewed measure of system success as reported in this study. The skewed measure probably accounts for the lack of findings when system success was considered as the dependent variable. This is perhaps the most severe limitation of this study.

However, we contend this type of study (i.e., using a survey) is more appropriate for this situation than the alternatives. The alternatives are: (1) develop several systems under the various conditions and test the success. This is not realistic given the already large application backlog. (2) Direct observations are not reasonable. With some systems having development times of several years, it would be impractical, if not impossible, to investigate several systems (Edstrom, 1977).

6.5 Suggestions for Further Research

Studies that are exploratory in nature are intended to detect and map out the main relationships in some sub-area of a discipline. In this context, this study serves as a guide, establishing research directions to be examined in the future, more refined investigations.

The most obvious suggestion for further research is to replicate this study, with a greater focus on the variables that have been identified as significant in this study. Also, other variables that were not suggested by the literature may affect the selection of a prototyping strategy and should be investigated. Thus, a replication and extension of this study is necessary.

More rigorous tests should be performed. This study has used a survey to collect the data. A more rigorous test, such as a lab experiment, would provide more definitive conclusions. Unfortunately, it is very difficult, without access to many different companies, to perform such research. Also, many companies are unwilling to allow lab studies due to the already existing backlogs of application systems and shortage of resources.

This study suffered from a range restriction in the measure of success. Further research is needed which will provide a more even distribution of successful and unsuccessful systems. As mentioned earlier, many of the relationships involving system success may not have been detected due to the small range in system success.

Another extension of this work would include a finer definition of prototyping. There are at least five types of prototyping strategies that have been identified (Doke, 1990) - this study has only used a very broad, three category approach (expendable, evolutionary, no prototyping). It would be interesting to investigate the various contingencies based upon the finer definition.

One of the contingencies proven in this study involved the use of tools for evolutionary prototyping. A continuation of this finding could include a look at the increasing use of evolutionary prototyping over time with the increase of sophisticated prototyping tools. We may very well find that the only thing restricting the full use of prototyping is the lack of tools (or lack of experience using the newest tools).

Finally, it was determined in this study that approximately 27 percent of respondents did not use prototyping and 73 percent did use prototyping. Further research should determine the decision variables used by organizations in determining the prototyping strategy used. For example, why did some organizations not use prototyping? Was it because of cost, or other factors? Also, why did some organizations use prototyping? The answers to these questions may prove a better guide to selecting a prototyping strategy.

6.6 Chapter VI Summary

This study has attempted to build a comprehensive model of prototyping strategy selection. Towards this end, data was gathered via a questionnaire from organizations throughout the country. A large representative sample was obtained.

Results from this study indicate that most of the suggestions offered in the literature as a way of selecting a prototyping strategy are not followed. Furthermore, the selection of a prototyping strategy, as suggested by the contingencies, did not seem to have an effect on system success.

The results suggest that prototyping is one of several factors that affect the success of a system. Prototyping, or any of the other factors, cannot be viewed in isolation. Instead, all of the factors, including prototyping, must be evaluated when developing an information system. Prototyping alone did not significantly affect system success.

The findings from this study should help shape future research in the specific area of prototyping and the broad area of systems development. For prototyping research, this study provides a better focus on future studies. Empirical evidence is now available to support, and dispute, the contingencies that have been accepted for many years. For systems development researchers, these results provide evidence that individual variables or small subsets of variables cannot be evaluated in isolation, rather all characteristics of an information system must be considered.

BIBLIOGRAPHY

Adamski, L. "The Prototyping Process," Systems International (UK), June 1985, 91-92.

Ahituv, N., Hadass, M., and Neumann, S., "A Flexible Approach to Information System Development," MIS Quarterly, 8 (2), June 1984, 69-78.

Alavi, M., "An Assessment Of The Prototyping Approach To Information Systems Development," Communications of the ACM, 27 (6), June 1984, 556-563.

Alavi, M., "The Evolution of Information Systems Development Approach: Some Field Observations," DATA BASE, 15 (3), Spring 1984, 19-24.

Alavi, M., and Henderson, J.C., "An Evolutionary Strategy for Implementing a Decision Support System," Management Science, 27, November 1981, 1309-1323.

Alavi, M., and Napier, A., "An Experiment in Applying the Adaptive Design Approach to DSS Development," Information and Management, 7, February 1984, 21-28.

Alreck, P.L., and Settle, R.B., The Survey Research Handbook, Irwin, Homewood, IL, 1985.

Alter, S., and Ginzberg, M., "Managing Uncertainty in MIS Implementation," Sloan Management Review, 20, Fall 1978, 23-31.

Alwin, Duane F., ed., Survey Design and Analysis: Current Issues, SAGE Publications, Beverly Hills, CA, 1978.

Andrews, William C., "Prototyping Information Systems," Journal of Systems Management, 34, September 1983, 16-18.

Asner, M., and King, A., "Prototyping: A Low-Risk Approach to Developing Complex Systems," Business Quarterly, 46 (3), Autumn 1981, 30-34.

Bailey, J.E., and Pearson, S.W., "Development of a Tool for Measuring and Analyzing Computer User Satisfaction," Management Science, 29 (5), May 1983, 530-545.

Baroudi, J.J., and Orlikowski, W.J., "A Short-Form Measure of User Information Satisfaction: A Psychometric Evaluation and Notes on Use," Journal of Management Information Systems, 4 (4), Spring 1988, 44-59.

Baroudi, J.J., and Orlikowski, W.J., "The Problem of Statistical Power in MIS Research," MIS Quarterly, 13 (1), March 1989, 87-106.

Benzecri, J.-P., Correspondence Analysis Handbook, Marcel Dekker, Inc., New York, NY, 1992.

Berrisford, T., and Wetherbe, J., "Heuristic Development: A Redesign of Systems Design," MIS Quarterly, 3 (1), March 1979, 11-19.

Boar, B., Application Prototyping: A Requirements Definition for the 80's. John Wiley & Sons, New York, NY, 1984.

Boar, B., "Application Prototyping: A Life Cycle Perspective," Journal of Systems Management, 24, February 1986, 25-31.

Boehm, B.W., Gray, T.E., and Seewaldt, T., "Prototyping Versus Specifying: A Multiproject Experiment," IEEE Transactions on Software Engineering, 10 (3), May 1984, 290-303.

Box, T.M., Performance Predictors for Entrepreneurial Manufacturing Firms: An Empirical Study of Psychological, Background and Environmental Scanning Attributes, unpublished Ph.D dissertation, Oklahoma State University, July 1991.

Brancheau, J.C., and Wetherbe, J.C., "Key Issues in Information Systems Management," MIS Quarterly, 11 (1), March 1987, 23-45.

Brittan, J.N.G., "Design for a Changing Environment," The Computer Journal, 23 (1), 1980, 13-19.

Brooks, F.P., "No Silver Bullets," Unix Review, November 1987, 39-48.

Brousseau, J., "Project Management: Look Before You Leap," Computing Canada, 14, April 28, 1988, 25.

Burns, R.N., and Dennis, A.R., "Selecting the Appropriate Application Development Methodology," DATA BASE, 17 (1), Fall 1985, 19-23.

Carey, J.M., "Prototyping: Alternative Systems Development Methodology," Information and Software Technology, 32 (2), March 1990, 119-126.

Carey, J.M., and Currey, J.D., "The Prototyping Conundrum," Datamation, 35 (11), June 1, 1989, 29-33.

Carey, J.M., and McLeod, R., "Use of System Development Methodology and Tools," Journal of Systems Management, 39 (3), March 1988, 30-35.

Carey, T.T., and Mason, R.E.A., "Information System Prototyping: Techniques, Tools, and Methodologies," INFOR, 21, August 1983, 177-191.

Cerveny, R.P., Garrity, E.J., Hunt, R.G., Kirs, P.J., Sanders, G.L., and Sipior, J.C., "Why Software Prototyping Works," Datamation, 33, August 1987, 97-98, 103.

Cerveny, R.P., Garrity, E.J., and Sanders, G.L., "The Application of Prototyping to Systems Development: A Rationale and Model," Journal of Management Information Systems, 3, 1986, 52-62.

Cerullo, M.J., "Information System Success Factors," Journal of Systems Management, 31, December 1980, 10-19.

Cohen, J., Statistical Power Analysis for the Behavioral Sciences, Academic Press, New York, NY, 1969.

Cohen, J., Statistical Power Analysis for the Behaviroal Sciences, 2nd ed., Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.

Cohen, J., "Things I Have Learned (So Far)," American Psychologist, 45 (12), December 1990, 1304-1312.

Cook, T.D., and Campbell, D.T., Quasi-Experimentation: Design and Analysis Issues for Field Studies, Houghton Mifflin Company, Boston, MA, 1979.

Connell, J.L., and Brice, L., "The Impact of Implementing a Rapid Prototype on System Maintenance," AFIPS Conference Proceedings, 1985, 515-522.

Connell, J., and Brice, L., "Rapid Prototyping," Datamation, 30, August 15, 1984, 93-100.

Cronbach, L.J., "Test Validation," in Educational Measurement, 2nd ed., R.L. Thorndike (ed.), American Council on Education, Washington, D.C., 1971, 443-507.

Davis, G.B., "Strategies for Information Requirements Determination," IBM Systems Journal, 21, 1982, 4-31.

Dearnley, P.A., and Mayhew, P.J., "In Favour of System Prototypes and Their Integration into the Systems Development Cycle," The Computer Journal, 26 (1), 1983, 36-42.

DeLone, W.H., and McLean, E.R., "Information Systems Success: The Quest for the Dependent Variable," Information Systems Research, 3 (1), March 1992, 60-95.

Dennis, A.R., Burns, R.N., and Gallupe, R.B., "Phased Design: A Mixed Methodology for Application System Development," DATA BASE, 18 (4), Summer 1987, 31-37.

Dickson, G.W., Leitheiser, R.L., Wetherbe, J.C., and Nechis, M., "Key Information Systems Issues for the 1980's," MIS Quarterly, 8 (3), September 1984, 135-159.

Doke, E.R., "An Industry Survey of Emerging Prototyping Methodologies," Information & Management, 18, April 1990, 169-176.

Doke, E.R., Hardgrave, B.C., and Swanson, N.E., "Prototyping and the Systems Development Life Cycle: An Investigation of the Relationships," 1991 Proceedings of the National Decision Sciences Institute, November 1991.

Doke, E.R., Swanson, N.E., and Hardgrave, B.C., "The Decision to Prototype Information Systems: A Pilot Study," National Proceedings of the Decision Sciences Institute, November 1992.

Doll, W.J., and Torkzadeh, G., "The Measurement of End-User Computing Satisfaction," MIS Quarterly, 12 (2), June 1988, 259-274.

Dos Santos, B.L., "A Management Approach to Systems Development Projects," Journal of Systems Management, 37, August 1986, 35-41.

Dos Santos, B.L., "MIS Project Management: A Contingency Approach," Management Decision, 26 (6), 1988, 22-28.

Dubin, R., Theory Building, The Free Press, New York, NY, 1969.

Eastman, K.K., The Effect of Situational Factors on Supervisor Perceptions of Potentially Ingratiating Behaviors: An Application of Kelley's Covariation Model, unpublished Ph.D dissertation, University of Nebraska, May 1990.

Edstrom, A., "User Influence and the Success of MIS Projects: A Contingency Approach," Human Relations, 30, 1977, 589-607.

El Louadi, M., Pollalis, Y.A., and Teng, J.T.C., "Selecting a Systems Development Methodology: A Contingency Framework," Information Resources Management Journal, 4 (1), Winter 1991, 11-19.

Everitt, B.S., The Analysis of Contingency Tables, Chapman and Hall, London, 1977.

Faris, R.E.L., ed., Handbook of Modern Sociology, Rand McNally & Company, Chicago, IL, 1964.

Fiedler, F.E., A Theory of Leadership Effectiveness, McGraw-Hill Book Company, New York, NY, 1967.

Fiedler, F.E., and Mahar, L., "A Field Experiment Validating Contingency Model of Leadership Training," Journal of Applied Psychology, 64 (3), 1979, 247-254.

Field, R.H.G., "A Critique of the Vroom-Yetton Contingency Model of Leadership Behavior," Academy of Management Review, 4, 1979, 249-257.

Franz, C.R., and Robey, D., "Organizational Context, User Involvement, and the Usefulness of Information Systems," Decision Sciences, 17, 1986, 329-355.

Freeman, D.H., Jr., Applied Categorical Data Analysis, Marcel Dekker, Inc., New York, NY, 1987.

Fowler, F.J., Jr., Survey Research Methods, Sage Publications, Newbury Park, CA, 1988.

Fuerst, W., and Cheney, P., "Factors Affecting the Perceived Utilization of Computer-Based Decision Support Systems in the Oil Industry," Decision Sciences, 13 (4), 1982, 554-569.

Galliers, R.D., "In Search of a Paradigm for Information Systems Research," in Research Methods in Information Systems, E. Mumford, et al., ed, Elsevier Science Publishers, Amsterdam, The Netherlands, 1985, 281-297.

Gavurin, S.L., "Where Does Prototyping Fit into IS Development?" Journal of Systems Management, 42, February 1991, 13-17.

Gibson, C.F., Singer, C.J., Schnidman, A.A., and Davenport, T.A., "Strategies for Making an Information Systems Fit Your Organization," Management Review, 73, January 1984, 8-14.

Gibson, H.L., and Rademacher, R.A., "The Top Ten Project Management Approaches," Data Management, 25 (3), March 1987, 21-25.

Gilhooley, I.A., "Productive Systems Development with Prototyping," Journal of Information Systems Management, 4, Spring 1987, 15-22.

Ginzberg, M.J., "A Study of the Implementation Process," TIMS Studies in the Management Sciences, R. Doktor, R.L. Schultz, and D.P. Slevin, eds., North-Holland Publishing Company, Volume 13, 1979, pp. 85-102.

Goodman, L.A., Analyzing Qualitative/Categorical Data, Abt Books, Cambridge, MA, 1978.

Graham, D.R., "Incremental Development: Review of Nonmonolithic Life-Cycle Development Models," Information & Software Technology, 31 (1), January/February 1989, 7-20.

Gremillion, L.L., and Pyburn, P., "Breaking the Systems Development Bottleneck," Harvard Business Review, 83, March-April 1983, 130-137.

Gronbaek, K., "Rapid Prototyping with Fourth Generation Systems - An Empirical Study," Office: Technology and People, 5 (2), 1989, 105-125.

Groner, G.F., Hopwood, M.D., Palley, N.A., and Sibley, W.L., "Requirements Analysis in Clinical Research Information Processing - A Case Study," Computer, 12 (9), September 1979, 108.

Guimaraes, T., "Prototyping: Orchestrating for Success," Datamation, 33 (23), December 1, 1987, 101-106.

Guimaraes, T., "Understanding Implementation Failure," Journal of Systems Management, 31, March 1981, 12-17.

Guimaraes, T., Igbaria, M., and Lu, M., "The Determinants of DSS Success: An Integrated Model," Decision Sciences, 23, 1992, 409-430.

Guimaraes, T., and Saraph, J.V., "The Role of Prototyping in Executive Decision Systems," Information & Management, 21, 1991, 257-267.

Gupta, Y.P., "Directions of Structured Approaches in System Development," Industrial Management & Data Systems (UK), July/August 1988, 11-18.

Gutierrez, O., "Experimental Techniques for Information Requirements Analysis," Information & Management, 16, 1989, 31-43.

Hair, J.F., Jr., Anderson, R.E., Tatham, R.L., and Black, W.C., Multivariate Data Analysis with Readings, 3rd ed., Macmillan Publishing Company, New York, NY, 1987.

Hardgrave, B.C., Doke, E.R., and Swanson, N.E., "Prototyping Effects on the System Development Life Cycle: An Empirical Study," Journal of Computer Information Systems, Spring 1993, forthcoming.

Harrison, R., "Prototyping and the Systems Development Life Cycle," Journal of Systems Management, 36, August 1985, 22-25.

Hartog, C., and Herbert, M., "1985 Opinion Survey of MIS Managers: Key Issues," MIS Quarterly, 10, December 1986, 351-361.

Huffaker, D., "Prototyping Business Applications Within the Traditional Life Cycle," Journal of Information Systems Management, 3, Fall 1986, 71-74.

Iivari, J., and Karjalainen, M., "The Impact of Prototyping on User Information Satisfaction During the IS Specification Phase," Information & Management, 17, 1989, 31-45.

Iivari, J., and Koskela, E., "The PIOCO Model for Information Systems Design," MIS Quarterly, 11 (3), September 1987, 401-419.

Ives, B., and Olson, M.H., "User Involvement and MIS Success: A Review of Research," Management Science, 30, 1984, 586-603.

Ives, B., Olson, M.H., and Baroudi, J.J., "The Measurement of User Information Satisfaction," Communications of the ACM, 26 (10), October 1983, 785-793.

Janson, M., Smith, L.D., "Prototyping for Systems Development: A Critical Appraisal," MIS Quarterly, 9, December 1985, 305-316.

Jenkins, M.A., Naumann, J.D., and Wetherbe, J.C., "Empirical Investigation of Systems Development Practices and Results," Information & Management, 7, 1984, 73-82.

Johnson, J.R., "A Prototypical Success Story," Datamation, 29 (10), November 1983, 251-252, 256.

Kanuk, L., and Berenson, C., "Mail Surveys and Response Rates: A Literature Review," Journal of Marketing Research, 12, November 1975, 440-453.

Kast, F.E., and Rosenzweig, J.E. Contingency Views of Organization and Management, Science Research Associates, Chicago, IL, 1973.

Kauber, P.G., "Prototyping: Not a Method But a Philosophy," Journal of Systems Management, 26, September 1985, 28-33.

Kerlinger, F.N., Foundations of Behavioral Research, 2nd ed., Holt, Rinehart and Winston, Inc., New York, NY, 1973.

Klingler, D.E., "Rapid Prototyping Revisited," Datamation, 32, October 1986, 131-132.

Klingler, D.E., "The Ten Commandments of Prototyping," Journal of Information Systems Management, Summer 1988, 66-72.

Kraemer, H.C., and Thiemann, S., How Many Subjects? Statistical Power Analysis in Research, SAGE Publications, Newbury Park, CA, 1987.

Kraushaar, J.M., and Shirland, L.E., "A Prototyping Method for Applications Development by End Users and Information Systems Specialists," MIS Quarterly, 9 (3), September 1985, 189-197.

Krzanik, L., "Anomalies with Evolutionary Innovation Project Control Strategies," in The Art and Science of Innovation Management, H. Hubner (ed.), Elsevier Science Publishers, Amsterdam, The Netherlands, 1986.

Langle, G.B., Leitheiser, R.L., and Naumann, J.D., "A Survey of Applications Systems Prototyping in Industry," Information & Management, 7, 1984, 273-284.

Larcker, D.F., and Lessig, V.P., "Perceived Usefulness of Information: A Psychometric Examination," Decision Sciences, 11 (1), 1980, 121-134.

Li, E.Y., "Software Testing in a System Development Process: A Life Cycle Perspective," Journal of Systems Management, 41 (8), August 1990, 23-31.

Lipp, M.E., "Working Conference on Prototyping," Information and Management, 7, 1984, 149-168.

Lucas, H.C., Jr., Why Information Systems Fail, Columbia University Press, New York, NY, 1975.

Luthans, F., Introduction to Management: A Contingency Approach. McGraw-Hill, New York, NY, 1976.

Lynch, R.M., "MIS Project Risk and Applications Development Approach," Industrial Management & Data Systems (UK), May/June 1987, 22-26.

Lyytinen, K., "Different Perspectives on Information Systems: Problems and Solutions," ACM Computing Surveys, 19 (1), March 1987, 5-46.

Lyytinen, K., "Expectation Failure Concept and Systems Analysts View of Information Systems Failures: Results of an Exploratory Study," Information and Management, 14, 1988, 45-56.

Mahmood, M.A., "System Development Methods - A Comparative Investigation," MIS Quarterly, 11, September 1987, 293-311.

Martin, E.W., "Strategy for a DoD software initiative," IEEE Computer, 16, March 1983, 52-59.

Mason, R.E.A., and Carey, T.T., "Prototyping Interactive Information Systems," Communications of the ACM, 26 (5), May 1983, 347-354.

McFarlan, F.W., "Portfolio Approach to Information Systems," Harvard Business Review, 59, September-October 1981, 142-150.

McFarlan, F.W., McKenney, J.L., and Pyburn, P., "The Information Archipelago - Plotting a Course," Harvard Business Review, 83 (1), January-February 1983, 145-156.

McKeen, J.D., "Successful Development Strategies for Business Application Systems," MIS Quarterly, 7, September 1983, 47-65.

McKeen, J.D., Naumann, J.D., and Davis, G.B., "Development of a Selection Model for Information Requirements Determination," Working Paper MISRC-WP-70-06, Graduate School of Business, University of Minnesota, 1979.

Meyer, K., and Kovacs, A., "Model Systems," Datamation, 29 (9), September 1983, 248-252.

Meyers, J.L., Fundamentals of Experimental Design, 3rd ed., Allyn and Bacon, Boston, MA, 1969.

Moore, J.H., "A Framework for MIS Software Development Projects," MIS Quarterly, 3, March 1979, 29-38.

Naumann, J.D., Davis, G.B., and McKeen, J.D., "Determining Information Requirements: A Contingency Method for Selection of a Requirements Assurance Strategy," Journal of Systems and Software, 1, 1980, 273-281.

Naumann, J.D., and Jenkins, A.M., "Prototyping: The New Paradigm for Systems Development," MIS Quarterly, 6 (3), September 1982, 29-44.

Necco, C.R., Gordon, C.L., and Tsai, N.W., "Systems Analysis and Design: Current Practices," MIS Quarterly, 11 (4), December 1987, 461-475.

Niederman, F., Brancheau, J.C., and Wetherbe, J.C., "Information Systems Management Issues for the 1990s," MIS Quarterly, 15 (4), December 1991, 475-500.

Nunnally, J., Psychometric Theory, 2nd ed., McGraw-Hill Publishing Company, New York, NY, 1978.

Olson, M.H., and Ives, B., "User Involvement in System Design: An Empirical Test of Alternative Approaches," Information and Management, 4, 1981, 183-195.

Oppenheim, A.N., Questionnaire Design and Attitude Measurement, Basic Books, Inc., New York, NY, 1966.

Orlikowski, W.J., and Baroudi, J.J., "Studying Information Technology in Organizations: Research Approaches and Assumptions," Information Systems Research, 2 (1), March 1991, 1-29.

Owen, D.E., "Prototyping: Essence of Pragmatic IS Development," Information Strategy: The Executive's Journal, Winter 1989, 21-25.

Palvia, P., and Nosek, J.T., "An Empirical Evaluation of System Development Methodologies," Information Resources Management Journal, 3 (3), Summer 1990, 23-32.

Pliskin, N., and Shoval, P., "End-User Prototyping: Sophisticated Users Supporting System Development, DATA BASE, 18 (4), Summer 1987, 7-17.

Pressman, R.S., Software Engineering: A Practitioner's Approach, 3rd ed., McGraw-Hill, New York, NY, 1992.

Rea, L.M., and Parker, R.A., Designing and Conducting Survey Research: A Comprehensive Guide, Jossey-Bass Publishers, San Francisco, CA, 1992.

Ryckman, H.O., "Requirement Definition Techniques," Journal of Information Management, 8 (3), Summer 1987, 17-21.

Saarinen, T., "System Development Methodology and Project Success," Information and Management, 19, 1990, 183-193.

Sanders, G.L., and Courtney, J.F., "A Field Study of Organizational Factors Influencing DSS Success," MIS Quarterly, 9 (1), March 1985, 77-93.

Schach, S.R., Software Engineering, Irwin, Homewood, IL, 1990.

Schoonhoven, C.B., "Problems with Contingency Theory: Testing Assumptions Hidden within the Language of Contingency 'Theory'," Administrative Science Quarterly, 26, 1981, 349-377.

Schonberger, R.J., "MIS Design: A Contingency Approach," MIS Quarterly, 4 (1), March 1980, 13-20.

Schroeder, D.L., A Prototype Management Support System: Design, Implementation, and Empirical Testing of a Natural Language-Like Knowledge-Based Interface Between an End User and a Statistical Problem Processing Package for Microcomputers, unpublished Ph.D dissertation, Oklahoma State University, July 1990.

Schultz, H., and Eierman, M., "Information Centers, Prototyping and Fourth Generation Languages," Data Management, 25 (4), April 1987, 26-31.

Schwartz, H., and Jacobs, J., Qualitative Sociology: A Method to the Madness, The Free Press, New York, NY, 1979.

Sethi, V., and King, W.R., "Construct Measurement in Information Systems Research: An Illustration in Strategic Systems," Decision Sciences, 22, 1991, 455-472.

Sethi, V., and Teng, J.T.C., "Choice of an Information Requirements Method: An Integrated Approach," INFOR, 26 (1), February 1988, 1-16.

Shah, A., An Inter-Industry Empirical Study Investigating the Relationship Between Environment and Marketing Strategy in Services, unpublished Ph.D dissertation, Oklahoma State University, May 1991.

Sharma, S., Durand, R.M., and Gur-arie, O., "Identification and Analysis of Moderator Variables," Journal of Marketing Research, 18, August 1981, 291-300.

Shoval, P., and Pliskin, N., "Structured Prototyping: Integrating Prototyping into Structured System

Development," Information & Management, 14 (1), 1988, 19-30.

Slusky, L., "Rapid Software Development Using Conceptual Modeling and Prototyping," Data Management, 25 (12), December 1987, 18-21.

Smith, R.B., "Project Life Cycles Aid the Systems Development Process," Data Management, 25 (3), (March 1987), 10-15, 41.

Srinivasan, A., and Kaiser, K.M., "Relationships Between Selected Organizational Factors and Systems Development," Communications of the ACM, 30 (6), June 1987, 556-562.

Sroka, J.M., and Rader, M.H., "Prototyping Increases Chance of Systems Acceptance," Data Management, 24 (3), March 1986, 12-19.

Straub, D.W., "Validating Instruments in MIS Research," MIS Quarterly, 13 (2), June 1989, 147-165.

Swanson, E.B., and Beath, C.M., "Departmentalization in Software Development and Maintenance," Communications of the ACM, 33, 1990, 658-667.

Swift, M.K., "Prototyping in IS Design and Development," Journal of Systems Management, 40, July 1989, 14-20.

Tait, P., and Vessey, I., "The Effect of User Involvement on System Success: A Contingency Approach," MIS Quarterly, 12 (1), March 1988, 91-108.

Tate, G., and Verner, J., "Case Study of Risk Management, Incremental Development, and Evolutionary Prototyping," Information and Software Technology, 32 (3), April 1990, 207-14.

Teng, J.T.C., and Galletta, D.F., "MIS Research Directions: A Survey of Researchers' Views," DATA BASE, Fall 1990, 1-10.

Teng, J.T.C., and Sethi, V., "A Comparison of Information Requirements Analysis Methods: An Experimental Study," DATA BASE, Winter 1990, 27-39.

Tillman, G., "Prototyping for the Right Results," Datamation, 35 (7), April 1, 1989, 42, 45.

Tozer, J.E., "Prototyping as a System Development Methodology: Opportunities and Pitfalls," Information and Software Technology, 29 (5), June 1987, 265-269.

Weill, P., and Olson, M.H., "An Assessment of the Contingency Theory of Management Information Systems," Journal of Management Information Systems, 6 (1), Summer 1989, 59-85.

Weinberg, R.S., "Prototyping and the Systems Development Life Cycle," Journal of Information Systems Management, Spring 1991, 47-53.

West, L.J., "A Frequently Violated Requirement in Survey Research," Journal of Education for Business, 66, January/February 1991, 134-135.

West, L.J., "Distinguishing Between Statistical and Practical Significance," Delta Pi Epsilon Journal, 32 (1), Winter 1990, 1-4.

West, L.J., "Sampling Techniques," Delta Pi Epsilon Journal, 5 (4), July 1963, 115-125.

Voltmer, J.G., "Selling Management on the Prototyping Approach," Journal of Systems Management, 40 (7), July 1989, 24-25.

Vroom, V.H., ed., Methods of Organizational Research, University of Pittsburgh Press, Pittsburgh, PA, 1967.

Vroom, V.H., and Jago, A.G., "On the Validity of the Vroom-Yetton Model," Journal of Applied Psychology, 63, 1978, 151-162.

Vroom, V.H., and Yetton, P.W., Leadership and Decision-Making, University of Pittsburgh Press, Pittsburgh, PA, 1973.

Yaverbaum, G.J., "Specifying System Requirements: A Framework of Current Techniques," Journal of Information Systems Management, Winter 1989, 17-21.

Zmud, R., Information Systems in Organizations, Scott, Foresman and Company, 1983.

APPENDIXES

APPENDIX A

PILOT STUDY

Oklahoma State University

COLLEGE OF BUSINESS ADMINISTRATION

STILLWATER, OKLAHOMA 74078-0555
BUSINESS 201
405-744-5064
FAX 405-744-5180

February 1, 1993

Chris Burton
Data-Tronics Corp.
P.O. Box 305
Fort Smith, AR 72902

Dear Mr. Burton:

A few weeks ago you were asked to consider participating in a study we are conducting concerning information systems in your organization. To reiterate, the purpose of this study is to investigate the characteristics of information systems and the development approach used. By collecting data on information systems in a variety of organizations, we can study how these characteristics influence the development approach and, subsequently, system success. The results of the study will provide guidelines for choosing the appropriate development approach based upon the characteristics of a system. We hope you have considered the importance of this study and are willing to participate.

We need you to select two information systems - one "good" system and one "bad" system - that have been implemented in your organization in the last two years. By "information system," we mean any system used in the normal course of your business, such as an accounting system. This definition would exclude such systems as spreadsheets and expert systems. For each of the two systems, a key developer and a primary user will need to complete a questionnaire. The questionnaires will only take a few minutes to complete, and postage-paid return envelopes are provided. On the first page of each questionnaire, please identify each system you select so the developers and users will know which system the questionnaire is addressing. Their comments should be directed only to the system you identify.

All responses are confidential. No individual organization or person will be identified. If you, the developer(s), or user(s) wish to see the results of this study, please provide your mailing information on the questionnaire. We deeply appreciate your participation in this study. Thank you in advance for your cooperation.

Sincerely,

Rick Wilson, Ph.D
Assistant Professor
Management Information Systems

Bill Hardgrave
Ph.D Candidate
Management Information Systems

Enclosures

This study is conducted by Bill Hardgrave, under the direction of Dr. Rick Wilson, as required for the completion of the doctoral dissertation in Management Information Systems at Oklahoma State University.

DEVELOPER QUESTIONNAIRE

The purpose of this study is to investigate the characteristics of information systems and the development approach used. As a system developer, your responses are very important. The information you provide will be used to develop guidelines for choosing the appropriate development approach based upon the characteristics of a system. The following pages contain questions concerning a recently implemented information system in your company. Please answer each question as accurately as possible. The questionnaire has been designed so that you can complete it very quickly and easily. All responses are confidential. No individual organization or person will be identified. Thank you.

Name, or ID, of system: ______________________________(provided by MIS manager)

Throughout the questionnaire, the term "system" refers to the system identified above.

SURVEY RESULTS

If you wish to receive a copy of the survey results, please provide your name and mailing address in the space provided below, or attach a business card. Your identity will be strictly confidential.

____

____

____

INDUSTRY DATA

Please provide the following information:

Your title: ______________________________

Years of experience: ______________________________

Your organization's major product(s) or service(s): ______________________________

Number of employees of organization: ______________________________

Number of information systems personnel: ______________________________

Please circle ONE response per question, or fill in the blank, unless otherwise noted.

A. DEVELOPER ATTRIBUTES

1. What was the size of the development team for this system? ____________

2. Prior to this system, how experienced was the development team in the system's application area?

1	2	3	4	5	6	7
no experience			understand domain, but no experience			extensive experience

3. Where are application programmers and analysts generally located?

1	2	3
user organizations	Centralized IS organization	other (specify) ____________

(please continue on the back of this page ...)

B. GENERAL SYSTEM ATTRIBUTES

1. General purpose of the system: ______________________________
____

2. How long has the system been in use? ______________________________

3. Is the system on-line or batch?

1	2
on-line	batch

** If batch, does the system have an interactive interface? ________

4. How much time, in months, was needed to develop the system (time to develop includes analysis, design, construction, testing, and implementation)? ______________________________

5. How much time was originally anticipated? ______________________________

6. How many total development man-hours did the system require (development includes analysis, design, construction, testing, and implementation)? ______________________________

7. Did the system represent a new development effort, or a modification (i.e., redesign, enhancement, etc.) to an existing system?

1	2	3	4	5	6	7
new development			major modification to existing system			minor modification to existing system

8. What was the original budget for the system? ________________

9. What is the annual budget of the IS department? ____________

10. What was the final cost of the completed system? ______________

11. How does this system rank, in cost, compared to other systems developed during the last two years?

1	2	3	4	5	6	7
upper 10%	upper 20%	upper 30%	about average	lower 30%	lower 20%	lower 10%

12. How does this system rank, in development time, compared to other systems developed during the last two years?

1	2	3	4	5	6	7
upper 10%	upper 20%	upper 30%	about average	lower 30%	lower 20%	lower 10%

C. MANAGEMENT and USER ATTRIBUTES

1. Prior to system development, top management felt that the time and resources needed for the development of this system was a wise investment ....

1	2	3	4	5	6	7
strongly disagree	moderately disagree	slightly disagree	neither agree or disagree	slightly agree	moderately agree	strongly agree

2. Was there a need for experimentation and learning before commitment of resources for a full system?

1	2
yes	no

3. How many users interacted with the development team during the development effort? __________

(please continue on the next page ...)

D. IMPORTANCE OF SYSTEM

1. Does the system operate, manage, and control the daily business activities of the organization?

1	2
yes	no

2. What would be the impact on the company if the system were to fail (or has failed)?

1	2	3	4	5	6	7
devastating			major impact			small impact

3. How would you describe the performance requirements of the system (e.g., response time, throughput)?

1	2	3	4	5	6	7
strict requirements			moderate requirements			low requirements

4. How would you describe the volume (number) of transactions through the system?

1	2	3	4	5	6	7
low			medium			high

5. What is the degree of usage of the system?

1	2	3	4	5	6	7
seldom used			frequently used			always used

6. How well defined was the process that the system supports?

1	2	3	4	5	6	7
very well defined			somewhat defined			not defined at all

7. Were the inputs and outputs required of the system specified in advance?

1	2	3	4	5	6	7
specified completely			partially specified			not specified

8. Did the system requirements change after development started?

1	2	3	4	5	6	7
no changes			some changes			many changes

E. USE OF PROTOTYPING

An information system prototype is a model of a system. A prototype can be as simple as mock-ups of reports or screens, or as complete as software that actually does some processing. Prototypes can be built with the intention of discarding them after they are no longer needed or they become part of the final operational system. Prototyping is the process of developing prototypes.

1. Using the above definition, has your organization used prototyping in the past? _______

If you answered "no" to the previous question, please skip to question 10 (on the next page).

2. Which ONE of the following statements describes the prototyping strategy used for this system?

    1. Only mock-ups of reports and screens were produced.
    2. Prototype simulates some system functions, but does not use real data.
    3. Prototype performs some actual system functions and uses real data.
    4. The prototype evolved into the final system.
    5. No form of prototyping was used.

(please continue on the back of this page ...)

3. What percentage of your organization's projects use prototyping?

1	2	3	4	5	6	7
0%	20%	40%	50%	60%	80%	100%

4. Does your organization use any of the following tools to build prototypes? (circle all that apply)

1. no tools   2. text editors/word processors   3. database management system
4. fourth generation language   5. other (please specify) ________________

5. Which of the following tools were used for this system? (circle all that apply)

1. no tools   2. text editors/word processors   3. database management system
4. fourth generation language   5. other (please specify) ________________

6. Which of the following tools have you used, prior to this system? (circle all that apply)

1. no tools   2. text editors/word processors   3. database management system
4. fourth generation language   5. other (please specify) ________________

7. In this organization, top management is strongly in favor of the concept of prototyping ....

1	2	3	4	5	6	7
strongly disagree	moderately disagree	slightly disagree	neither agree or disagree	slightly agree	moderately agree	strongly agree

8. Prior to this system, were explicit procedures established for planning and controlling the project?

1	2
yes	no

9. Please list the factors that you feel should be considered in the decision whether or not to use prototyping for a specific system development project. For each factor listed, indicate the importance of the factor on a scale from 1 to 10 (1=low importance, 10=high importance).

________________ | ________________
________________ | ________________
________________ | ________________
________________ | ________________
________________ | ________________
________________ | ________________

10. How experienced are you with the use of prototypes?

1	2	3	4	5	6	7
no experience			understand concept but no experience			extensive experience

11. Does your organization plan to use prototyping in the future?

1	2
yes	no (why not? ________________)

THANK YOU FOR YOUR TIME AND ASSISTANCE!

USER QUESTIONNAIRE

The purpose of this study is to investigate the characteristics of information systems and the development approach used. As a user, your responses are quite important. The information you provide will be used to develop guidelines for choosing the appropriate development approach based upon the characteristics of a system. Results of this study can assist developers in providing information systems to you, the user, in a more timely manner that better meets your needs. The following pages contain questions concerning a recently implemented information system in your organization. Please answer each question as accurately as possible. The questionnaire has been designed so that you can complete it very quickly and easily. All responses are confidential. No individual organization or person will be identified. Thank you.

Name, or ID, of system: ______________________________ (provided by MIS manager)

Throughout the questionnaire, the term "system" refers to the system identified above. "User" refers to members of the company that use the system.

SURVEY RESULTS

If you wish to receive a copy of the survey results, please provide your name and mailing address in the space provided below, or attach a business card. Your identity will be strictly confidential.

____

____

____

Please circle ONE response per question, or fill in the blank, unless otherwise noted.

1. Your title: ______________________________

2. Prior to this system, how much experience did you have in using computerized systems?

1	2	3	4	5	6	7
no experience			limited experience			extensive experience

3. How many users regularly use the system? ______________

4. Did the system change the way you performed your job?

1	2	3	4	5	6	7
no changes			some changes			many changes

5. Did user departments have to reorganize to meet the requirements of the system?

1	2	3	4	5	6	7
no reorganization			some reorganization			major reorganization

6. Users are often asked to work with the developers of a system by specifying the requirements for the system. In terms of your participation with the developers of this system, you have ....

1	2	3	4	5	6	7
participated to to a great extent			participated to to a moderate extent			did not participate

If you answered "did not participate" to question 6, please skip to question 9 (on the next page).

(please continue on the back of this page ...)

7. Were you freed from your normal duties during times of participation, or was the participation time added to your normal workload?

1	2
freed from normal duties	participation was added to workload

8. On average, how many hours per week were spent with the developers of the system? ____________

9. Not including this system, have you ever been involved in a system development effort?

1	2	3	4	5	6	7
never involved			previous involvement but limited knowledge			high degree of involvement

10. An information system prototype is a model of a system. A prototype can be as simple as mock-ups of reports or screens, or as complete as software that actually does some processing. Prototypes can be built with the intention of discarding them after they are no longer needed or they become part of the final operational system. Prototyping is the process of developing prototypes. Prior to this system, how knowledgeable were you with the concept of "prototyping?"

1	2	3	4	5	6	7
no knowledge			limited knowledge			extensive knowledge

11. Did you learn about prototyping during the development of this system?

1	2
yes	no

12. How closely does the system match what you wanted and expected from the system?

1	2	3	4	5	6	7
more than expected			satisfactory match			poor match

13. Do you believe the originally stated objectives for the system were satisfied?

1	2	3	4	5	6	7
definitely			not certain			definitely not

14. Have there been implementation problems associated with this system in your organization?

1	2	3	4	5	6	7
no problems			moderate problems			very serious problems

15. How well has this system been accepted by your organization?

1	2	3	4	5	6	7
enthusias-tically			satisfactory acceptance			very negatively

16. How would you rate your satisfaction with this system?

1	2	3	4	5	6	7
extremely low	quite low	slightly low	neither high or low	slightly high	quite high	extremely high

(please continue on the next page ...)

17. Please indicate the degree of congruence between what you wanted or required and what is provided by the information system (please circle one response from each of the following two scales).

1	2	3	4	5	6	7
extremely useful	quite useful	slightly useful	neither useful or useless	slightly useless	quite useless	extremely useless

1	2	3	4	5	6	7
extremely relevant	quite relevant	slightly relevant	neither relevant or irrelevant	slightly irrelevant	quite irrelevant	extremely irrelevant

18. Please indicate the correctness of the output from the information system (please circle one response from each of the following two scales).

1	2	3	4	5	6	7
extremely inaccurate	quite inaccurate	slightly inaccurate	neither accurate or inaccurate	slightly accurate	quite accurate	extremely accurate

1	2	3	4	5	6	7
extremely low	quite low	slightly low	neither high or low	slightly high	quite high	extremely high

19. Please indicate the consistency and dependability of the information from the system (please circle one response from each of the following two scales).

1	2	3	4	5	6	7
extremely high	quite high	slightly high	neither high or low	slightly low	quite low	extremely low

1	2	3	4	5	6	7
extremely superior	quite superior	slightly superior	neither superior or inferior	slightly inferior	quite inferior	extremely inferior

20. Please indicate the precision of the output from the system (please circle one response from each of the following two scales).

1	2	3	4	5	6	7
extremely high	quite high	slightly high	neither high or low	slightly low	quite low	extremely low

1	2	3	4	5	6	7
extremely definite	quite definite	slightly definite	neither definite or uncertain	slightly uncertain	quite uncertain	extremely uncertain

---

THANK YOU FOR YOUR TIME AND ASSISTANCE!

APPENDIX B

DATA COLLECTION LETTERS AND QUESTIONNAIRES

Oklahoma State University

COLLEGE OF BUSINESS ADMINISTRATION

STILLWATER, OKLAHOMA 74078-0555
BUSINESS 201
405-744-5064
FAX 405-744-5180

February 26, 1993

Dan Moylan
United Van Lines
One United Dr.
Fenton, MO 63026

Dear Dan Moylan:

The topic of software development has received much attention in recent years as organizations are forced to find ways to improve the development process. As a member of the information systems community, you have been selected to participate in a survey regarding this topic.

In the next few days you will be asked to respond to a mail survey concerning information systems used in your organization. The purpose of this study is to investigate the characteristics of information systems and the development approach used. The results of the study will provide guidelines for choosing the appropriate development approach based upon the characteristics of a system. A proper match between the development approach and system characteristics can increase the likelihood of successful development and implementation of a system.

Your cooperation is extremely important. To collect this information, we need data regarding information systems used by various organizations. We will be asking you to select two information systems implemented within your organization in the past two years. One of the systems should be a "good" system, the other a "bad" system. For each of the two systems, we will provide a questionnaire for a key developer and a primary user of the system. The questionnaires will only take a few minutes to complete, and postage-paid return envelopes will be provided.

At the completion of this study, we will be happy to share the results with you. Thank you for your consideration.

Sincerely,

Rick Wilson, Ph.D
Assistant Professor
Management Information Systems
(405) 744-5084

Bill Hardgrave
Ph.D Candidate
Management Information Systems
(405) 744-8620

This study is conducted by Bill Hardgrave, under the direction of Dr. Rick Wilson, as required for the completion of the doctoral dissertation in Management Information Systems at Oklahoma State University.

Oklahoma State University

COLLEGE OF BUSINESS ADMINISTRATION

STILLWATER, OKLAHOMA 74078-0555
BUSINESS 201
405-744-5064
FAX 405-744-5180

March 4, 1993

Dan Moylan
United Van Lines
One United Dr.
Fenton, MO 63026

Dear Dan Moylan:

A few days ago you should have received a letter asking you to participate in a study we are conducting concerning information systems in your organization. We hope you have considered the importance of this study and are willing to participate.

By collecting data on information systems in a variety of organizations, we can study how these characteristics influence the development approach and, subsequently, system success. As stated in the earlier letter, we need you to select two information systems - one "good" system and one "bad" system - that have been implemented in your organization in the *last two years*. By "information system," we mean any system used in the normal course of your business, such as an accounting system. This definition would exclude such systems as spreadsheets and expert systems. For each of the two systems, a key developer and a primary user will need to complete a questionnaire. The questionnaires will only take a few minutes to complete, and postage-paid return envelopes are provided. *On the first page of each questionnaire, please identify each system you select so the developers and users will know which system the questionnaire is addressing.* Their comments should be directed only to the system you identify.

All responses are confidential. No individual organization or person will be identified. If you, the developer(s), or user(s) wish to see the results of this study, please provide your mailing information on the questionnaire. We deeply appreciate your participation in this study. Thank you in advance for your cooperation.

Sincerely,

Rick Wilson, Ph.D
Assistant Professor
Management Information Systems
(405) 744-5084

Bill Hardgrave
Ph.D Candidate
Management Information Systems
(405) 744-8620

Enclosures

This study is conducted by Bill Hardgrave, under the direction of Dr. Rick Wilson, as required for the completion of the doctoral dissertation in Management Information Systems at Oklahoma State University.

DEVELOPER QUESTIONNAIRE

The purpose of this study is to investigate the characteristics of information systems and the development approach used. As a system developer, your responses are very important. The information you provide will be used to develop guidelines for choosing the appropriate development approach based upon the characteristics of a system. The following pages contain questions concerning a recently implemented information system in your company. Please answer each question as accurately as possible. The questionnaire has been designed so that you can complete it very quickly and easily. All responses are confidential. No individual organization or person will be identified. Thank you.

Name, or ID, of system: ______________________________(provided by MIS manager)

Throughout the questionnaire, the term "system" refers to the system identified above.

Please circle ONE response per question, or fill in the blank, unless otherwise noted.

A. INDUSTRY DATA

1. Your title: ______________________________

2. Years of experience: ______________________________

3. Your organization's major product(s) or service(s): ______________________________

4. Number of employees of organization: ______________________________

5. Number of information systems personnel: ______________________________

B. DEVELOPER ATTRIBUTES

1. What was the size of the development team for this system? ____________

2. Prior to this system, how experienced was the development team in the system's application area?

1	2	3	4	5	6	7
no experience			understand domain, but no experience			extensive experience

3. Where are application programmers and analysts generally located?

1	2	3
user organizations	Centralized IS organization	other (specify) ____________

C. GENERAL SYSTEM ATTRIBUTES

1. General purpose of the system: ______________________________

____

2. How long has the system been in use? ______________________________

(please continue on the back of this page ...)

3. Is the system on-line or batch?

1	2	3
on-line	batch	elements of both

** If batch, does the system have an interactive interface? ________

4. How much time, in months, was needed to develop the system (time to develop includes analysis, design, construction, testing, and implementation)? ________

5. How much time (to develop) was originally anticipated? ________

6. How many total development man-hours did the system require (development includes analysis, design, construction, testing, and implementation)? ________

7. Did the system represent a new development effort, or a modification (i.e., redesign, enhancement, etc.) to an existing system?

1	2	3	4	5	6	7
new development			major modification to existing system			minor modification to existing system

8. What was the original budget for the system? ________

9. What is the annual budget of the Information Systems department? ________

10. What was the final cost of the completed system? ________

11. How does this system rank, in cost, compared to other systems developed during the last two years?

1	2	3	4	5	6	7
upper 10%	upper 20%	upper 30%	about average	lower 30%	lower 20%	lower 10%

12. How does this system rank, in development time, compared to other systems developed during the last two years?

1	2	3	4	5	6	7
upper 10%	upper 20%	upper 30%	about average	lower 30%	lower 20%	lower 10%

13. How many systems have been implemented in your organization during the last two years? ________

D. MANAGEMENT and USER ATTRIBUTES

1. Prior to system development, top management felt that the time and resources needed for the development of this system was a wise investment ....

1	2	3	4	5	6	7
strongly disagree	moderately disagree	slightly disagree	neither agree or disagree	slightly agree	moderately agree	strongly agree

2. Was there a need for experimentation and learning before commitment of resources for a full system?

1	2
yes	no

3. How many users interacted with the development team during the development effort? ________

4. On average, how many hours per week did developers spend with the users during development (development includes analysis, design, construction, testing, and implementation)? ________

(please continue on the next page ...)

E. IMPORTANCE OF SYSTEM

1. Does the system operate, manage, and control the daily business activities of the organization?

1	2
yes	no

2. What would be the impact on the company if the system were to fail (or has failed)?

1	2	3	4	5	6	7
devastating			major impact			small impact

3. How would you describe the performance requirements of the system (e.g., response time, throughput)?

1	2	3	4	5	6	7
strict requirements			moderate requirements			low requirements

4. How would you describe the volume (number) of transactions through the system?

1	2	3	4	5	6	7
low			medium			high

5. What is the degree of usage of the system?

1	2	3	4	5	6	7
seldom used			frequently used			always used

6. How well defined was the process that the system supports?

1	2	3	4	5	6	7
very well defined			somewhat defined			not defined at all

7. Were the inputs and outputs required of the system specified in advance?

1	2	3	4	5	6	7
specified completely			partially specified			not specified

8. Did the system requirements change after development started?

1	2	3	4	5	6	7
no changes			some changes			many changes

9. Overall, how would you rate the users' satisfaction with this system?

1	2	3	4	5	6	7
extremely low	quite low	slightly low	neither high or low	slightly high	quite high	extremely high

F. USE OF PROTOTYPING

Definition: An information system prototype is a model of a system. A prototype can be as simple as mock-ups of reports or screens, or as complete as software that actually does some processing. Prototypes can be built with the intention of discarding them after they are no longer needed or they become part of the final operational system. Prototyping is the process of developing prototypes.

1. Using the above definition, has your organization used prototyping in the past? _______

If you answered "no" to the previous question, please skip the next page and go to question 10.

(please continue on the back of this page ...)

2. Which ONE of the following statements describes the prototyping strategy used for this system?

A. Only mock-ups of reports and screens were produced.
B. Prototype simulates some system functions, but does not use real data.
C. Prototype performs some actual system functions and uses real data.
D. The prototype evolved into the final system.
E. No form of prototyping was used.

3. What percentage of your organization's projects use ...
(scale = 0 to 100%)

A. Only mock-ups of reports and screens. ______
B. Prototype simulates some system functions, but does not use real data. ______
C. Prototype performs some actual system functions and uses real data. ______
D. The prototype evolved into the final system. ______
E. No form of prototyping. ______

4. Does your organization use any of the following tools to build prototypes? (circle all that apply)

1. no tools 2. text editors/word processors 3. database management system
4. fourth generation language 5. other (please specify) ______________________

5. Which of the following tools were used for this system? (circle all that apply)

1. no tools 2. text editors/word processors 3. database management system
4. fourth generation language 5. other (please specify) ______________________

6. Which of the following tools have you used, prior to this system? (circle all that apply)

1. no tools 2. text editors/word processors 3. database management system
4. fourth generation language 5. other (please specify) ______________________

7. In this organization, top management is strongly in favor of the concept of prototyping ....

1	2	3	4	5	6	7
strongly disagree	moderately disagree	slightly disagree	neither agree or disagree	slightly agree	moderately agree	strongly agree

8. Prior to this system, were explicit procedures established for planning and controlling the project?

1	2
yes	no

9. Please list the factors that you feel should be considered in the decision whether or not to use prototyping for a specific system development project. For each factor listed, indicate the importance of the factor on a scale from 1 to 10 (1=low importance, 10=high importance).

______________________ | ______________________
______________________ | ______________________
______________________ | ______________________
______________________ | ______________________
______________________ | ______________________
______________________ | ______________________

(please continue on the next page ...)

10. How experienced are you with the use of prototypes?

1	2	3	4	5	6	7
no experience			understand concept but no experience			extensive experience

11. Does your organization plan to use prototyping in the future?

1 yes

2 no (why not? ______________________________)

THANK YOU FOR YOUR TIME AND ASSISTANCE!

Please return this questionnaire by March 26, 1993, or at your earliest convenience.

- -

cut here

SURVEY RESULTS

If you wish to receive a copy of the survey results, please provide your name and mailing address in the space provided below. Return this section with the questionnaire, or, if you wish, remove this section at the dashed line above and return separately to: Bill Hardgrave, Department of Management, College of Business Administration, Oklahoma State University, Stillwater, OK, 74078. Your identity will be strictly confidential.

____

____

____

USER QUESTIONNAIRE

The purpose of this study is to investigate the characteristics of information systems and the development approach used. As a user, your responses are quite important. The information you provide will be used to develop guidelines for choosing the appropriate development approach based upon the characteristics of a system. Results of this study can assist developers in providing information systems to you, the user, in a more timely manner that better meets your needs. The following pages contain questions concerning a recently implemented information system in your organization. Please answer each question as accurately as possible. The questionnaire has been designed so that you can complete it very quickly and easily. All responses are confidential. No individual organization or person will be identified. Thank you.

Name, or ID, of system: ______________________________ (provided by MIS manager)

Throughout the questionnaire, the term "system" refers to the system identified above. "User" refers to members of the company that use the system.

Please circle ONE response per question, or fill in the blank, unless otherwise noted.

---

1. Your title: ____

2. Prior to this system, how much experience did you have in using computerized systems?

1	2	3	4	5	6	7
no experience			limited experience			extensive experience

3. How many users regularly use the system? ______________

4. Did the system change the way you performed your job?

1	2	3	4	5	6	7
no changes			some changes			many changes

5. Did user departments have to reorganize to meet the requirements of the system?

1	2	3	4	5	6	7
no reorganization			some reorganization			major reorganization

6. Users are often asked to work with the developers of a system by specifying the requirements for the system. In terms of your participation with the developers of this system, you have ....

1	2	3	4	5	6	7
participated to to a great extent			participated to to a moderate extent			did not participate

If you answered "did not participate" to question 6, please skip to question 9 (on the next page).

7. Were you freed from your normal duties during times of participation, or was the participation time added to your normal workload?

1	2
freed from normal duties	participation was added to workload

(please continue on the back of this page ...)

8. On average, how many hours per week did you spend with the developers of the system? ____________

9. Not including this system, have you ever been involved in a system development effort?

1	2	3	4	5	6	7
never involved			previous involvement but limited knowledge			high degree of involvement

10. An information system prototype is a model of a system. A prototype can be as simple as mock-ups of reports or screens, or as complete as software that actually does some processing. Prototypes can be built with the intention of discarding them after they are no longer needed or they become part of the final operational system. Prototyping is the process of developing prototypes. Prior to this system, how knowledgeable were you with the concept of "prototyping?"

1	2	3	4	5	6	7
no knowledge			limited knowledge			extensive knowledge

11. Did you learn about prototyping during the development of this system?

1	2
yes	no

12. How closely does the system match what you wanted and expected from the system?

1	2	3	4	5	6	7
more than expected			satisfactory match			poor match

13. Do you believe the originally stated objectives for the system were satisfied?

1	2	3	4	5	6	7
definitely			not certain			definitely not

14. Have there been implementation problems associated with this system in your organization?

1	2	3	4	5	6	7
no problems			moderate problems			very serious problems

15. How well has this system been accepted by your organization?

1	2	3	4	5	6	7
enthusias-tically			satisfactory acceptance			very negatively

16. How would you rate your satisfaction with this system?

1	2	3	4	5	6	7
extremely low	quite low	slightly low	neither high or low	slightly high	quite high	extremely high

17. Please indicate the degree of congruence between what you wanted or required and what is provided by the information system (please circle one response from each of the following two scales).

1	2	3	4	5	6	7
extremely useful	quite useful	slightly useful	neither useful or useless	slightly useless	quite useless	extremely useless

1	2	3	4	5	6	7
extremely relevant	quite relevant	slightly relevant	neither relevant or irrelevant	slightly irrelevant	quite irrelevant	extremely irrelevant

(please continue on the next page ...)

18. Please indicate the correctness of the output from the information system (please circle one response from each of the following two scales).

1	2	3	4	5	6	7
extremely inaccurate	quite inaccurate	slightly inaccurate	neither accurate or inaccurate	slightly accurate	quite accurate	extremely accurate

1	2	3	4	5	6	7
extremely low	quite low	slightly low	neither high or low	slightly high	quite high	extremely high

19. Please indicate the consistency and dependability of the information from the system (please circle one response from each of the following two scales).

1	2	3	4	5	6	7
extremely high	quite high	slightly high	neither high or low	slightly low	quite low	extremely low

1	2	3	4	5	6	7
extremely superior	quite superior	slightly superior	neither superior or inferior	slightly inferior	quite inferior	extremely inferior

20. Please indicate the precision of the output from the system (please circle one response from each of the following two scales).

1	2	3	4	5	6	7
extremely high	quite high	slightly high	neither high or low	slightly low	quite low	extremely low

1	2	3	4	5	6	7
extremely definite	quite definite	slightly definite	neither definite or uncertain	slightly uncertain	quite uncertain	extremely uncertain

---

THANK YOU FOR YOUR TIME AND ASSISTANCE!

Please return this questionnaire by March 26, 1993, or at your earliest convenience.

- -

cut here

SURVEY RESULTS

If you wish to receive a copy of the survey results, please provide your name and mailing address in the space provided below. Return this section with the questionnaire, or, if you wish, remove this section at the dashed line above and return separately to: Bill Hardgrave, Department of Management, College of Business Administration, Oklahoma State University, Stillwater, OK, 74078. Your identity will be strictly confidential.

____

____

____

____

March 22, 1993

Dear Dan Moylan:

A few weeks ago you should have received a mail survey containing two sets of questionnaires titled "Developer Questionnaire" and "User Questionnaire."

If you have decided to participate in this study, I would like to express my appreciation for your contribution. However, if you have not yet participated, I would like to urge you to do so. Since only a few individuals from your industry have been selected to take part in this study, your input is of great importance. Would you please take a few minutes of your time to help with this study? Thank you again for your assistance.

Sincerely,

Bill

Bill Hardgrave

Oklahoma State University

OSU

Oklahoma State University

COLLEGE OF BUSINESS ADMINISTRATION

STILLWATER, OKLAHOMA 74078-0555
BUSINESS 201
405-744-5064
FAX 405-744-5180

April 7, 1993

Dan Moylan
United Van Lines
One United Dr.
Fenton, MO 63026

Dear Dan Moylan:

A few weeks ago you were asked to respond to a mail survey concerning information systems used in your organization. If you have responded, we would like to express our appreciation for your contribution. In the event you haven't responded, would you please consider doing so? Your participation will only require a few minutes, and the aggregate results can be very useful to you and others in the Information Systems industry. In case you have misplaced the previous survey, we have enclosed a replacement set of questionnaires.

To participate, we need you to select an information system that has been implemented in your organization in the past two years. On the first page of each of the enclosed questionnaires, please identify the system you have selected. Next, for the selected system, dispense the questionnaires to a developer and a user. The questionnaires will only take a few minutes to complete, and postage-paid return envelopes are provided. All responses are strictly confidential. No individual organization or individual will be identified.

Since only a few individuals from your industry have been selected to take part in this study, *your participation is very important*. Please help us with this study. Thank you again for your assistance.

Sincerely,

Bill Hardgrave
Ph.D Candidate
Management Information Systems

Rick Wilson, Ph.D
Assistant Professor
Management Information Systems

Enclosures

This study is conducted by Bill Hardgrave, under the direction of Dr. Rick Wilson, as required for the completion of the doctoral dissertation in Management Information Systems at Oklahoma State University.

2

VITA

Billy Charles Hardgrave

Candidate for the Degree of

Doctor of Philosophy

Thesis: A CONTINGENCY MODEL FOR SELECTING AN INFORMATION SYSTEM PROTOTYPING STRATEGY

Major Field: Business Administration

Area of Specialization: Management Information Systems

Biographical:

Personal Data: Born in Clarksville, Arkansas, January 20, 1964, the son of Glen N. and Evelyn Hardgrave.

Education: Graduated from Hartman High School, Hartman, Arkansas, in May 1982; received the Bachelor of Science Degree in Computer Science from Arkansas Tech University in December 1987 (with honors); received the Master of Business Administration at Southwest Missouri State University in August 1990; completed the requirements for the Doctor of Philosophy at Oklahoma State University in July 1993.

Professional Experience: Programmer/Analyst, Computec, Inc., June 1985 to May 1986; General Manager, Ellis Software, Inc., May 1986 to June 1989; Graduate Assistant, Southwest Missouri State, June 1989 to August 1989 and January 1990 to August 1990; Lecturer, Southwest Missouri State, September 1989 to December 1989; Graduate Teaching Associate, Oklahoma State University, August 1990 to June 1993.

Honors and Awards: 1993 Phoenix Outstanding Doctoral Student Award - OSU Graduate College; 1992 College of Business "Faculty of the Month" (April) - Business Student Council; 1992 Faculty Appreciation Award - Office of Greek Life; 1991 Outstanding Graduate Teaching Award - Golden Key National Honor Society.