

## INFORMATION TO USERS

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

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

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

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

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

# U·M·I

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



**AN INVESTIGATION INTO THE ORGANIZATIONAL AND  
TECHNOLOGICAL FACTORS THAT CONTRIBUTE TO THE  
SUCCESSFUL IMPLEMENTATION OF CASE TECHNOLOGY**

by

**CONSTANCE A. KNAPP**

A dissertation submitted to the Graduate Faculty in Business in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1995

**UMI Number: 9530890**

**Copyright 1995 by  
Knapp, Constance Anne  
All rights reserved.**

---

**UMI Microform 9530890  
Copyright 1995, by UMI Company. All rights reserved.**

**This microform edition is protected against unauthorized  
copying under Title 17, United States Code.**

---

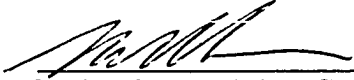
**UMI**

**300 North Zeeb Road  
Ann Arbor, MI 48103**

© 1995  
CONSTANCE ANNE KNAPP  
All Rights Reserved

This manuscript has been read and accepted for the Graduate Faculty in Business in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

4-6-95  
Date

  
Chair of Examining Committee

April 10, 1995  
Date

  
Executive Officer

Dr. David Dannenbring

Dr. George Schneller

Dr. George Sphicas

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

## Abstract

# AN INVESTIGATION INTO THE ORGANIZATIONAL AND TECHNOLOGICAL FACTORS THAT CONTRIBUTE TO THE SUCCESSFUL IMPLEMENTATION OF CASE TECHNOLOGY

by

Constance A. Knapp

Advisor: Professor M. Barry Dumas

This study examines the organizational and technological factors that contribute to the successful implementation of CASE (computer-aided software engineering) technology. A questionnaire was developed to determine a profile of CASE users. This profile was used to determine with whom interviews would be conducted. Data were collected through personal interviews with integrated CASE tool users. These interviews were analyzed using a grounded theory approach. A theory of CASE implementation was developed based on this analysis.

This study found that CASE implementation success relies on the interaction between management's understanding of information technology, the information systems development environment, and the complexity of application systems that are developed in an organization. Some of the factors underlying these core themes were suggested by the literature, but most emerged from the analysis of the data.

The core theme of organizational understanding of information systems technology is described by: the presence of a champion; the factors that were considered in the decision; the commitment of management; the expected benefits of CASE technology; and the role of an information systems development methodology in the organization. The core theme of

information systems development environment is described by: the skill set of the information systems professionals that the organization employs; the way the CASE tool is used; the implementation strategy chosen; and the role of an information systems development methodology in the organization. Finally, the core theme of information systems development complexity is described by: the training approach followed; the expected benefits of CASE technology; the implementation strategy used; and the role of an information systems development methodology in the organization. This study found that adherence to a systems development methodology is of particular importance when integrated CASE technology is being implemented.

This research adds to the body of knowledge that explores the relationship between factors in an organization and CASE adoption success. This work also extends the stream of information systems research which utilizes qualitative techniques.

## ACKNOWLEDGMENTS

I would like to express my gratitude to my committee, Dr. David Dannenbring, Dr. George Schneller and Dr. George Sphicas, for their support. I would especially like to thank Dr. Barry Dumas, the chair, for his encouragement, advice and unswerving confidence in me.

I would also like to thank the members of the Advisory Board of the School of Computer Science and Information Systems of Pace University for their time, their advice and most of all their willingness to participate in this research, either directly or indirectly. Thanks also are extended to all who were interviewed for this research.

I am also grateful to my fellow doctoral students, some of whom were in dissertation support groups with me, and some of whom listened and critiqued my work in our dissertation seminar. I am particularly grateful to Subhasish Dasgupta, Anthony Ioannidis and Phoebe Massimino.

I owe Dr. Susan Merritt, Dean, Pace University School of Computer Science and Information Systems, Professor Myron Goldberg, former chair of the Information Systems Department, and Dr. John Molluzzo, the current chair, a debt of gratitude for their support, including a semester of doctoral leave. They were willing to hear me say "no" to tasks when they would have preferred I say "yes" while I labored to complete this dissertation. My colleagues Professor Robert Benjes, Dr. Linda Jo Calloway and Dr. Melvyn Weisel of Pace University, and Dr. Virginia Fenton of Kean College were especially helpful.

Finally, thanks to Dr. Anne Corey, for her support during these difficult six and one half years. She led the way and then stood behind me. For her steadfast support and complete confidence in my ability to complete this dissertation, I am both amazed and grateful.

## Table of Contents

<b>Chapter One: Introduction.....</b>	<b>1</b>
<b>Overview.....</b>	<b>1</b>
<b>CASE Technology Overview and Definition.....</b>	<b>2</b>
<b>Description of This Research.....</b>	<b>5</b>
<b>Importance of This Research .....</b>	<b>8</b>
<b>Findings of This Research .....</b>	<b>9</b>
<b>Chapter Two: Literature Review.....</b>	<b>11</b>
<b>Introduction.....</b>	<b>11</b>
<b>Systems Development Life Cycle Research.....</b>	<b>11</b>
<b>Systems Development Methodology Research.....</b>	<b>18</b>
<b>Computer-Aided Software Engineering Research.....</b>	<b>24</b>
<b>Grounded Theory.....</b>	<b>30</b>
<b>Conclusions.....</b>	<b>33</b>
<b>Chapter Three: Research Methodology and Design.....</b>	<b>34</b>
<b>Introduction.....</b>	<b>34</b>
<b>Research Question .....</b>	<b>34</b>
<b>CASE Implementation Success .....</b>	<b>36</b>
<b>Classes of Factors .....</b>	<b>37</b>
<b>Research Methodology .....</b>	<b>40</b>
<b>Conclusions.....</b>	<b>45</b>
<b>Chapter Four: Data Analysis and Research Results .....</b>	<b>46</b>
<b>Introduction.....</b>	<b>46</b>
<b>CASE User Profile .....</b>	<b>46</b>
<b>The Use of Grounded Theory .....</b>	<b>50</b>
<b>Core Themes.....</b>	<b>54</b>
<b>A Theory of CASE Implementation.....</b>	<b>72</b>

<b>Chapter Five: Conclusions and Suggestions for Future Research.....</b>	<b>81</b>
<b>Conclusions.....</b>	<b>81</b>
<b>Implications for IS Practitioners .....</b>	<b>82</b>
<b>Implications for IS Researchers.....</b>	<b>84</b>
<b>Suggestions for Future Research .....</b>	<b>85</b>
<b>Appendix A: The Principles of Grounded Theory.....</b>	<b>86</b>
<b>Appendix B: The Questionnaire .....</b>	<b>93</b>
<b>Appendix C: Analysis of Quantitative Data.....</b>	<b>97</b>
<b>Appendix D: Interview Protocol .....</b>	<b>104</b>
<b>Appendix E: Samples of Code Notes.....</b>	<b>107</b>
<b>Reference List.....</b>	<b>110</b>

## List of Tables

Table 1.....	76
Table 2.....	77
Table 3.....	78
Table 4.....	103

List of Figures

Figure 1 ..... 79  
Figure 2 ..... 80

## Chapter One: Introduction

### Overview

American firms spent nearly \$1 billion in 1991 on CASE technology, the latest year for which data are available, yet not all of these firms consider their implementation of CASE technology to be successful (Weinman, 1992). This research identifies the organizational and technological factors that contribute to successful implementation of Computer-Aided Software Engineering (CASE) technology.

CASE technology refers to the software that assists information systems analysts and designers in the development of application software. CASE technology encompasses many kinds of CASE products. Some products are designed to accomplish the tasks of one phase of the information systems development life cycle. These tools are called CCASE, or component CASE tools. Other CASE products provide integration across the information systems development life cycle, and are commonly called ICASE tools. The shared repository, or encyclopedia, provided by these ICASE tools will eventually lead an organization to realize enhanced productivity and increased reusability of system components. The promise of improved productivity has led information systems managers to invest heavily in both kinds of CASE technology.

Despite the significant level of investment in CASE tools, no previous research has considered whether an organization that was successful in implementing ICASE technology had a particular profile. ICASE tools are of interest because this class of tools has, and will continue to have, a greater impact on information systems development than CCASE tools.

**This research attempts to answer the question are there certain characteristics of an organization that will predict whether ICASE will be successful in that organization?** In this dissertation organizations using ICASE tools were studied to learn what organizational and technological factors contributed to their success, or lack of success.

This dissertation is divided into five chapters: introduction, literature review, research methodology and design, data analysis and research results, and conclusions and suggestions for future research.

The introductory chapter includes a general description of CASE technology and a discussion of CASE implementation. In this chapter, CASE and its role in information systems development are defined. The research methods used are discussed, including the assumptions that were made and the scope of the study. Chapter one concludes with a discussion of the contribution of this research to information systems.

#### CASE Technology Overview and Definition

Information systems are often developed following some information systems development life cycle approach, involving an analysis of the problem, the design of a solution to the problem, the development of software, a test of software, and the implementation of the system. While there is no uniform agreement on a standard information systems development life cycle, researchers have found that an information systems development life cycle approach is widely used.

Systems analysts and designers use graphic tools, such as data flow diagrams, to depict information system requirements and design application systems. These diagrams can be drawn and maintained manually, but substantial effort is involved in checking them for correctness and maintaining them over the life of the information system. Initially,

software packages known as "analysts' workbenches" were developed to assist systems analysts in the development and maintenance of diagrams. Code generators were also available to help automate the writing of program source code. Tools that allowed an analyst to input a decision table as parameters for code generation were available in the 1970's. As these tools matured, they became more integrated. An analyst drawing data flow diagrams might also include functional specifications in structured English for some of the processes. These functional specifications would allow a programmer, or another analyst, to generate program source code using a code generator that might accept the output of the analysis program as its input.

A class of tools called CASE tools emerged in the early 1980's. These software tools were based on the earlier code generators and analysts' workbenches. There are many definitions of CASE. One of the earliest definitions of CASE defines CASE as "a system of automated software life cycle support tools that permits the generally accepted principles of software engineering to be effectively used in a practical and coordinated manner" (Souza, 1990). Software engineering principles include a system life cycle methodology, standard guidelines and procedures, structured methods and tools, productivity improvement techniques, and quality assurance and control.

ICASE tools depend on the use of a particular methodology. In this country most ICASE tools are based on information engineering. Often the tasks of one phase of the life cycle depend on the information developed during a previous phase. Integrated tools were designed with the expectation that tasks for all phases would be completed using the tool, according to the conventions of the methodology supported by the tool. If

the tool is used in this way a repository, or encyclopedia, is developed and shared between all phases of the information systems development life cycle. This repository contains all data elements, process names, descriptions, and all other information related to the application and is accessed by members of the project team during the various phases of information systems development. Because of the tight integration between life cycle phases, ICASE tools lack the flexibility of CCASE tools.

CCASE tools lack the integration of ICASE tools. In practice this often means that when CCASE tools are used for information systems development, the diagrams that were developed during the analysis phase aren't available for use in the design phase, and often must be at least partly regenerated. However, if tools are only being used for some activities during one phase of information systems development, such as generating data flow diagrams, a shared repository may not be necessary.

CASE technology assists organizations in meeting a growing backlog of software development projects. Besides developing new software applications, firms are maintaining previously developed applications. Software, unlike other assets, doesn't wear out and continues to be used. Many firms that developed applications in the 1960's continue to use them, making changes as needed to ensure that the software will run on current hardware, that the software meets current operational requirements, and that the applications perform as well as possible. Often more money is spent maintaining old systems than is spent developing new ones. CASE technology is being used to help "re-engineer" these older systems. This effort includes developing a repository to assist maintenance programmers in locating data elements and analyzing existing program source code and generating "structured" code (program source code that conforms to the

principles of structured programming). The use of CASE tools will change the way information systems are both developed and maintained by providing better documentation of analysis and design decisions, and by generating program source code that is easier to maintain.

Organizations now have approximately ten years of experience using CASE tools. While some organizations embraced the early tools with high expectations, users of CASE today better understand both the benefits and the limitations of this technology. CASE tools can be useful, both in developing new systems and re-engineering existing systems. However, the adoption of CASE technology will not necessarily solve all information systems problems. As this research shows, in those organizations where managers have realistic expectations of what CASE technology can accomplish, and are willing to invest both time and money in its adoption, CASE technology implementation can be successful. In those organizations where training needs have been underestimated, where the complexity of CASE technology implementation has been misunderstood, and where information systems are not developed under the discipline of an information systems development life cycle, the obstacles to success are much greater.

### Description of This Research

#### Methodology

**This research is based on in-depth interviews with current and past users of ICASE tools.** These users were identified in various ways. ICASE vendors were asked for names of CASE users. Participants were also solicited through electronic mail to an international CASE user's forum. Questionnaires were mailed to members of CASE users groups. These questionnaires were used to develop a preliminary profile of a

typical CASE user. Face-to-face or telephone interviews were conducted with participants who matched this preliminary profile.

The data collected were examined using qualitative analysis techniques. A theory of successful ICASE implementation was developed based on these data by following the principles of "grounded theory." The result is an explanation of an organization's success or lack of success with CASE technology. Grounded theory is a qualitative analysis approach that allows the data to "speak," through rigorous analysis and coding of concepts expressed in interviews. Grounded theory was first developed in sociology and has since been used in many other disciplines, including management and information systems research. It is invaluable in fields like information systems where theoretical research is being conducted. A fuller explanation of how grounded theory was used is provided in chapter four. Appendix A discusses the principles of grounded theory in more depth.

The concepts discovered in the interviews were classified based on the information systems literature. Categories were then developed to classify these concepts. These categories were integrated around a set of three core themes. In this way an explanation was developed based on the data collected to identify the factors that contribute to an organization's successful implementation of CASE technology.

A preliminary questionnaire was used to identify types of users to interview. The data from the questionnaire provided information about how long companies have been using the tools, the types of projects the tools are being used for, and how closely the organization adheres to an information systems development life cycle. The questionnaire provided

data on how successful, or satisfied, the respondents believed that their firm was with their particular implementation of CASE technology.

#### Scope of this research

This research focuses on ICASE tools. Of all CASE tools, ICASE tools provide the greatest benefit to an organization that is developing application systems because of their completeness and integration. ICASE tools share the following characteristics: they allow for the development and maintenance of a repository that is shared across all phases of the life cycle, they provide tools to complete the tasks of all phases of the life cycle, and they provide a way of expressing and storing the firm's business rules. While all ICASE tools are functionally similar, the mechanics of using the tools are slightly different. The layout of the menus, the names of various tasks and the specific steps followed to prepare a diagram differ from tool to tool. Tools differ in charting conventions. Some tools enforce an information systems development methodology more rigorously than others.

The focus of this study is the organizations that have adopted the technology, and not on the characteristics of particular products or vendors. Since ICASE tools are based on similar principles of software engineering, these tools are more like each other than they are different from each other.

ICASE tools require a large investment in software, and are commonly used only for large projects in large organizations. The experiences of these organizations may be different from the experience of smaller organizations taking advantage of other forms of CASE technology. The results of this research might not be generalizable to all CASE users.

The data were collected through in-depth interviews. In this way questions could be pursued that would be too open-ended for a questionnaire. As with all research that relies on self-reported data, this study assumes that the respondents are able to assess characteristics about the organization, and to report on them accurately. Qualitative analysis techniques were used to analyze these data. The objective was to discover those factors which contribute to an organization's successful use of CASE tools.

#### Importance of This Research

**This study identifies organizational and technological factors that contribute to the successful implementation of CASE technology, and examines the interaction among and between these factors.** No previous empirical work has examined this interaction. Much of the empirical work on CASE that has been reported assumes that CASE tools provide value, and that all organizations would benefit from using them, with little regard for organizational characteristics that might predispose success or lack of success. Organizations may realize some benefit from the use of CASE technology, but only if they are successful in implementing this technology. Success stories appear in the literature, along with the suggestion that if other organizations do what these organizations have done, they too will be successful. A variation of this approach is to report on what went wrong, and suggest what the organization might have done differently.

Researchers have studied the use of an information systems development life cycle. Some researchers have advocated the systems development life cycle approach while others have proposed a prototyping approach. Other researchers have compared and contrasted systems

development methodologies, such as information engineering and structured systems analysis and design. Researchers have also proposed ways to implement CASE technology in organizations. Some of these researchers have linked the success of CASE implementation with the successful implementation of systems in general. CASE technology incorporates both a life cycle and a methodology, which makes implementation of CASE different from implementation of either one alone. Since CASE supports information systems development, implementing CASE technology is also different from implementing information systems in general. CASE is not just another application system. CASE is the tool-kit that allows the development of application systems.

#### Findings of This Research

**This study found that CASE implementation success relies on the interaction between management's understanding of information technology, the information systems development environment, and the complexity of application systems that are developed in an organization.** Some of the factors underlying these core themes were suggested by the literature, but most emerged from the analysis of the data.

The core theme of organizational understanding of information systems technology is described by: the presence of a champion; the factors that were considered in the decision; the commitment of management; the expected benefits of CASE technology; and the role of an information systems development methodology in the organization. The core theme of information systems development environment is described by: the skill set of the information systems professionals that the organization employs; the

way the CASE tool is used; the implementation strategy chosen; and the role of an information systems development methodology in the organization. Finally, the core theme of information systems development complexity is described by: the training approach followed; the expected benefits of CASE technology; the implementation strategy used; and the role of an information systems development methodology in the organization.

This research adds to the body of knowledge that explores the relationship between factors in an organization and CASE adoption success. This exploratory study builds on the work of Orlikowski, Ryan and Beck, and Rowe (Orlikowski, 1988; Orlikowski, 1989; Orlikowski, 1988; Orlikowski, 1993; Orlikowski & Baroudi, 1991; Rowe, 1993; Ryan & Bock, 1992).. In addition, this work supports a direction of research that has been suggested by Wynekoop and Conger (Wynekoop & Conger, 1990). Finally, this work extends the stream of information systems research which utilizes qualitative techniques (Calloway, 1988; Calloway & Arlav, 1990; Fenton, 1992; Lee, 1989; Lee, 1991; Martin & Turner, 1986; Orlikowski, 1993).

The next chapter discusses the literature review. Systems development life cycle research, systems development methodology research, and computer-aided software engineering research were reviewed. In addition, previous studies using grounded theory were sought out and examined.

Chapter three describes the research methodology and design, and chapter four discusses the data analysis and research results. The final chapter presents conclusions and suggestions for future research.

## Chapter Two: Literature Review

### Introduction

The information systems development literature in general, and computer-aided software engineering (CASE) technology specifically, is reviewed in this chapter. Research is classified as theoretical or empirical, and major prior contributions in systems development life cycle research, systems development methodology research, and CASE technology research are discussed. Four classes of organizational and technological factors that are suggested by the literature are identified. In addition to classifying the literature, weaknesses in previous research are also identified. Finally, the theoretical development of grounded theory is described and introduced as a data analysis approach in information systems research.

### Systems Development Life Cycle Research

#### Overview

While there may be disagreement about the numbers of phases in the life cycle, previous research suggests that most organizations developing large systems use a life cycle approach (Necco, Gordon, & Tsai, 1987). The theoretical research in this area has consisted of comparing the systems development life cycle approach with a prototyping approach or integrating the two approaches. Most of the empirical work consists of surveys of users to determine what approaches are being used, and by whom.

#### Theoretical Research

Much of the theoretical research prescribes that a systems development life cycle approach should be used, the prototyping approach

should be used, or a contingency approach should be used, taking into account the characteristics of the project and the organization.

The information systems development life cycle is generally well accepted. However, critics argue that the life cycle is too rigid, requiring knowledge in the analysis phase that simply is not available (see for example Kraushaar & Shirland, 1985; Mahmood, 1987). Some researchers have suggested modifications that would make the systems development life cycle more flexible and iterative. For example, researchers have suggested that the life cycle should be adapted as necessary, depending on the environment and the nature of the development process (Ahituv, Hadass, & Neumann, 1984). Such research assists the practitioner in adapting the information systems development life cycle.

Incremental development, or the use of a non-monolithic life-cycle, is an alternative approach. While the discipline of a life cycle approach is needed, the inflexibility of carrying out the steps contributes to the problems information systems professionals have in delivering high quality systems on time (Graham, 1989). For example, in reviewing a number of approaches, Graham asserts that incremental development and incremental delivery will prove more successful than monolithic life cycle approaches, although he laments the dearth of reported experiences with incremental approaches.

Other researchers have argued that prototyping is a better way to develop systems. Prototyping should be used in combination with the more traditional systems development life cycle approach (Janson & Smith, 1985). Parallels can be drawn between software engineering and other types of engineering, and prototyping is widely used in other engineering systems design. Research conducted using a state transition model shows

that where system requirements are not well defined, prototyping has the potential to provide increased productivity for many types of information systems projects (Kraushaar & Shirland, 1985). Prototyping has been called the "new paradigm for systems development" and some argue that this approach should be widely used, supplanting the life cycle approach. "Structured problems require it" (Naumann & Jenkins, 1982, p. 42).

Many researchers see a blending of the two approaches as the solution to the problem of rigidity within the traditional systems development life cycle approach. One way to ensure the needed flexibility is to incorporate prototyping in the analysis and design phase of the systems development life cycle (Gavurin, 1991). Using a prototype in the analysis phase will increase the quality of the delivered system as well as shorten the time spent in the analysis phase. The use of a mixed methodology incorporating prototyping and the systems development life cycle has also been suggested (Dennis, Burns, & Gallupe, 1987; Weinberg, 1991). This method is sometimes called "phased design" because following the analysis phase the system is broken into a set of sub-systems, each of which is designed, coded and implemented.

In summary, most of the theoretical research discusses ways in which the traditional systems development life cycle model might be made more flexible, either by adapting the phases based on characteristics of the environment and the project being developed, or by incorporating prototyping into the life cycle. A few researchers have argued that the life cycle should be abandoned in favor of the quicker, more flexible, prototyping approach.

### Empirical Research

Most empirical work in this area has been devoted to determining how the systems development life cycle is applied in practice. The underlying assumption is that the information systems development life cycle approach is being used. What is studied is how the approach is applied. The research method that has been most widely used in systems development life cycle research is the field study. One such study focused on thirty-two application systems at random from five organizations. The purpose of this research was to determine the relationship between time spent in various phases of the systems development life cycle and the outcome of the development (McKeen, 1983). In particular, typical business application systems were studied. Only systems that had passed through multiple phases of development qualified for inclusion in this study. For each project chosen, the project leader and the primary user provided questionnaire data. Most of time spent on the system was spent in coding, far exceeding what would normally be recommended in the literature. Those projects that spent more time in the analysis phase were more successful, regardless of the outcome measure used. Practitioners should spend more time in the analysis phase of the information systems development life cycle. This research only studied projects that were developed using an information systems development life cycle.

Another survey of 72 information systems development projects in 23 major United States corporations was conducted (Jenkins, Naumann, & Wetherbe, 1984). Systems development managers were interviewed. The focus of this research was on the use of a systems development methodology. However, 44% of the project leaders viewed the systems development methodology as primarily a scheduler (as opposed to a

methodology that prescribes specific tools and techniques). Given that response, it is reasonable to conclude that the term "systems development methodology" was really being interpreted as "systems development life cycle." Most of the large organizations surveyed used some form of a systems development life cycle to manage and control projects.

When Gordon, Necco and Tsai (Gordon, Necco, & Tsai, 1987) described the systems development life cycle as "the process of breaking a computer-based system project into phases," they found that about two-thirds of the organizations responding to their questionnaire indicated that they used a systems development life cycle of some form<sup>1</sup>. These researchers also found that the time allocated to any phase varies widely, supporting earlier findings (McKeen, 1983). They were attempting to develop a "standard" life cycle, but found that life cycle phases and activities differ widely among the organizations surveyed. In fact, 16 different systems development life cycles were reported, each containing an analysis and general design phase, and a detailed design phase. The use of the investigation, installation, review and maintenance phases differed among the organizations. These researchers concluded that the systems development life cycle is merely a set of guidelines and not a "fixed methodology" (Gordon, et al., 1987, p. 27). This broader interpretation of a life cycle, in contrast to a methodology, is important. Most ICASE tools require the use of a methodology. The importance of this requirement will be discussed in the next section of this chapter.

A survey of Korean companies was conducted to determine the relationship between procedural formalization, which was defined as

---

1. This article reports on the same survey as (Necco, et al., 1987) but with a slightly different emphasis.

adhering to a formal information systems development life cycle, and MIS success (Lee & Kim, 1992). A formal development process and success were shown to be highly correlated, with a correlation coefficient of .44. This was not influenced by the organization's size, age, nor the structuredness of the task, and was true for all phases except installation and conversion. The authors suggest that this may be a function of Korean companies, which may have less experience with MIS than other developed countries. However, their work supports earlier findings (Jenkins, et al., 1984).

A survey of the Fortune 1000 found that firms are experimenting with a variety of prototyping methods (Doke, 1990). Four separate prototyping methodologies could be identified: illustrative, simulated, functional and evolutionary. An illustrative approach is used to illustrate possible designs, a simulated methodology uses simulation to present a model of the system, a functional methodology focuses on building models which provide system functions (without all the interfaces in place), and an evolutionary methodology starts with a small model and adds functions and interfaces over time. Over 60% of the respondent firms were using prototyping, and all four methodologies are being used. Prototyping is important in information systems development.

In an interesting attempt to measure the impact of a development method, in this case prototyping and the systems development life cycle, on a system's success, a regression model using data collected by questionnaires from data processing/management information systems managers and users of 54 systems was developed (Mahmood, 1987). This research found that the success of a development method is dependent upon outside factors such as a particular project's characteristics, such as project

size, cost and duration. A framework of these characteristics was developed, along with a six point scale for each, that will allow practitioners to determine which method to use.

The empirical research has found that the life cycle approach is used, albeit inconsistently in terms of the allocation of resources to each of the phases. Past research has supported the theoretical notion that systems developed using the discipline of a life cycle approach will be more successful than those developed with less discipline. However, dissatisfaction with the time required to analyze and design systems has given rise to experimentation with prototyping.

Further research needs to be done studying systems being built by non-traditional information systems staff, such as end user development. While end-user computing is beyond the scope of this study, one result of the perceived dissatisfaction with information systems departments may be a rise in end-user computing (Dickson, Leitheiser, & Wetherbe, 1984; Kraushaar & Shirland, 1985). Prototyping has proven very successful in development of "ad hoc" systems as well, although one of the disadvantages of using the approach is the lack of documentation of system requirements and design decisions (Davis, 1988). The interest in adapting the information systems development life cycle may be a result of the diffusion of systems development throughout the organization, so that this activity now includes those who are neither accustomed to the discipline required to follow a prescribed life cycle, nor appreciative of its benefits. A weakness of most of this research a failure to investigate how those firms who are not following a life cycle develop information systems, or why firms do not follow the prescribed approach.

## Systems Development Methodology Research

### Overview

Information systems professionals use tools and techniques, usually termed a "methodology," to develop application systems. This is true whether a life cycle approach is followed or not. Methodologies vary across cultures. This research focuses on methodologies in use in the United States. Both the theoretical and the empirical research emphasize comparing methodologies, sometimes to determine which is better, but often to provide practitioners with some guidelines to allow them to make more informed choices between methodologies.

Some of the theoretical literature merely presents the tools and techniques. Some researchers advocate a particular methodology. Most of the empirical literature either surveys practitioners to determine what is used, or compares and contrasts methodologies.

### Theoretical Research

Some of the earliest published work was aimed at practitioners. Nearly all of this initial work described the tools and techniques for structured analysis and design (DeMarco, 1978; Gane & Sarson, 1979; Yourdon, 1989; Yourdon & Constantine, 1979). Information engineering was introduced slightly later, although still around the same time (Martin, 1990a; Martin, 1990b; Martin, 1990c; Martin & Finkelstein, 1981). These books served mainly as textbooks for practitioners. Later, as the techniques were being blamed for the failure of information systems development (Davis, Lee, Nickles, Chatterjee, Hartung, & Wu, 1992; Kydd, 1989; Lederer & Nath, 1991), some of the developers of these techniques took the position of advocates, arguing that structured techniques remained important, and that it was failure to follow the

techniques, not the techniques themselves, that caused systems development activities to fail (Yourdon, 1986; Yourdon, 1989; Yourdon, 1990; Yourdon, 1991).

Some researchers argue that the forces of change in information systems are too strong to allow adherence to one, rigid methodology. Instead, we should be looking at blending methodologies. An example of a such a methodology is MultiView (Avison & Wood-Harper, 1991). The future of information systems development may not lie in adherence to a single method, but the ability to adapt methods as needed. Lyytinen states that "the tyranny of one method is gone" (Lyytinen, 1989, p. 9).

Finally, we need a clearer understanding of the assumptions that lie at the heart of any approach to systems development. Four paradigms of information system development have been suggested, each with its own assumptions about how knowledge is acquired and the nature of the world (Hirschheim & Klein, 1989). In each of these four paradigms, the analyst is seen as having a different role, either as expert, facilitator, labor partisan or emancipator. The functionalism paradigm with analyst as expert is the one that is exemplified by information engineering and structured analysis, according to the authors. It is time for recognition of the other roles that an analyst might play, even though, "... (i)n practice, information systems development approaches are influenced by assumptions from more than one paradigm. However, the influence from one paradigm is typically dominant" (Hirschheim & Klein, 1989, p. 1212). Recognition of these paradigms, and their underlying assumptions, might help explain unexpected outcomes in information systems projects.

In summary, the theoretical research into systems development methodologies has primarily focused on the development of particular

methods, such as structured analysis and design, or information engineering. Other researchers have proposed blending or adapting methodologies as required by the situation at hand. Additional attention should be focused on the implicit assumptions that designers make about the way knowledge is acquired and the world in which systems are developed.

### Empirical Research

Empirical research has consisted of surveys to determine what techniques and tools are being used, or comparisons carried out, either in the form of experiments using different methods, or in the development and application of a set of "rules" to allow comparisons to be made between methods.

Field research has included the development of a layered, or tiered model of application systems development, supported by interviews with systems developers (Curtis, Krasner, & Iscoe, 1988). This research focused on determining how systems development decisions were made on large development projects. Among other findings, the authors concluded that software development methods must be flexible enough to accommodate change, must become a medium of communication, and must be evaluated considering software development as a "learning, communication and negotiation process" (Curtis, et al., 1988). They do not advocate a single method, but a recognition of the fact that software development is more than the application of methods. Software development includes many domains of knowledge, including application knowledge, which tools and techniques do not address.

While some look at the behavioral aspects of systems development, others survey users to see what tools and techniques were being applied to systems development problems (Gordon, et al., 1987; Kievit & Martin,

1989; Necco, et al., 1987). One survey attempted to determine how organizations perform analysis and design. Using a questionnaire completed by computer executives, data was collected on approaches, documentation aids used in conjunction with the traditional/classical approach, techniques used in structured analysis and structured design, and the use of automated tools. Six approaches were defined: systems development life cycle, traditional/classical, structured, automated, prototyping, and information center. The purpose of this research was to determine how organizations perform analysis and design. These researchers found that organizations used more than one method, and that regardless of method most organizations felt that they were able to meet the user requirements, and develop systems on time and within budget.

A later survey was limited to tools and techniques, although the researchers don't specify how "tools and techniques" were defined (Kievit & Martin, 1989). A clear definition would have improved this work. Data dictionaries, prototyping, structured walk-throughs and design teams were classified as "planning methods tools." Data flow diagrams, system flow charts, decision tables, Warnier-Orr diagrams, Nassi-Schneiderman diagrams, hierarchical charts and IPO charts were classified as "design charting techniques." These tools serve different purposes, and grouping them together is a weakness of this research. For example, data flow diagrams focus on processes and the data used by these processes, while decision tables focus on decision rules. These tools are also used in analysis and not only in design. Prototypes are used not only to plan, but also to design interfaces for systems. Data flow diagrams and system flow charts were reported as the most frequently used charting tools, implying that

structured techniques and more traditional techniques co-existed at the time of this data collection.

Some researchers have compared methods, while others have compared tools within a method. One study compared structured analysis techniques to traditional methods (Colter, 1984). The researcher concluded that "no single tool, technique or methodology can support the complete analysis of today's complex systems" (Colter, 1984, p. 64). This conclusion contradicts the underlying philosophy of some ICASE tool developers, who design tools to fit only one methodology. In another situation, an experiment was conducted using graduate students assigned four week projects (Yadav, Bravoco, Chatfield, & Rajkumar, 1988). The purpose of the experiment was to compare, empirically, the success of data flow diagrams, DFD, and the integration definition method, IDEF. The authors chose these two techniques as representative of widely used methods. IDEFo, the function modeling part of IDEF, was used in the experiment. The authors extended the earlier framework along syntactic, semantic, communication ability and usability dimensions. This research was limited by the use of small projects, completed by 20 graduate students. The experimental results are inconclusive as to which method is better.

As information engineering gains in prominence, tools for using the methodology become more important. Information engineering is an approach to systems development that can be broadly defined as an approach that is "directed specifically at translating a corporate focus (a strategic plan, expressed as an organization mission statement) into an information systems architecture (ISA), which can be directly translated into data, application and geographic architectures" (Hackathorn & Karimi,

1988, p. 203). Using this broad definition, a framework for evaluating tools and techniques to accomplish this task has been proposed by Hackathorn and Karimi. While concluding that no tool, technique or methodology can cover the full life cycle, the researchers nevertheless believe that if information engineering methods provided ways to establish linkages between organizational strategies and information systems requirements, productivity could be increased at all levels of the organization. Because of the broad information engineering definition these researchers used, their conclusions apply to all types of systems development activities. This research also supports the use of information engineering as an underlying methodology for CASE tools. Many ICASE developers have based their tools on the principles of information engineering.

A comparison of Jackson System Development with real-time structured analysis as implemented in CASE/Real-time method was conducted. The Jackson System Development method is widely used in Europe, while CASE/RT is an extension of Yourdon methodology (Cameron, Campbell, & Ward, 1991). Of interest here is not the two methodologies as much as the methods used to compare them. The authors propose comparing methods along three dimensions: the notations, the rules for applying the notations, and the results produced by the method. They illustrate this approach by comparing the above mentioned methodologies. They found that "a method is not wholly characterized by its notations, that different looking notations can be used to capture the same information,...and that mathematically equivalent notations may each handle certain classes of problem better than others" (Cameron, et al., 1991, p. 402). They conclude that a fair degree of compatibility is possible

between and among methods, and that if carefully analyzed, few differences would be found. This conclusion implies that the use of a methodology is more important than the particular methodology chosen.

In summary, most systems development methodology research that is empirically based has focused on either examining what is used or comparing methods. No one particular method has been found to accomplish all the required tasks, nor are practitioners found to be focusing on one method. Instead, "the analyst must bring a set of tools to a particular analysis task and utilize those approaches that are necessary to produce a complete result" (Colter, 1984, p. 64). This is certainly no simple task, and is made even harder when the tools must be manually applied—data flow diagrams must be drawn and redrawn, and data dictionaries must be manually updated. Initial interest in automated tools was clearly driven by the fact that "people become frustrated with the amount of manual labor that was required to develop structured analysis models" (Yourdon, 1986, p. 133). However, CASE tools in general, and ICASE tools specifically, are usually based on either one particular method or a set of techniques. This research implies that while one particular method shouldn't be adhered to religiously, some method must be used for information systems development success.

### Computer-Aided Software Engineering Research

#### Overview

Computer-aided software engineering (CASE) research is relatively new. According to Wynekoop and Conger, "The number of papers discussing the topic grew from tens of articles in the early 1980's to hundreds in the last two years" (Wynekoop & Conger, 1990, p. 130). The exponential growth in this research, however, is of uneven quality. Many

of the papers relate unsupported prescriptive advice. Others report on a specific success or failure. Still others dogmatically support a particular product, methodology, or approach.

In this section significant CASE research is discussed. This review includes articles that appeared in the major information systems journals, journals devoted to CASE (e.g., CASE Trends), conference proceedings, and IEEE publications. Significant articles from trade publications such as Datamation and Computerworld were included where appropriate. Working papers from universities, doctoral and master's theses and other unpublished works were also sought. These sources were chosen to ensure that articles of primary importance were considered.

Theoretical literature consists mostly of normative writings. Empirical literature includes surveys, experiments and case studies. There is a dearth of literature empirically assessing how CASE technology is used. A notable exception is the work of Orlikowski (Orlikowski, 1988; Orlikowski, 1989; Orlikowski, 1988; Orlikowski, 1993; Orlikowski & Baroudi, 1991).

### Theoretical Research

Case (1986) wrote the earliest book on computer-aided software engineering. His was the first of many texts prescribing software tools to support information systems development (see for example Gane, 1990; McClure, 1989). The implicit assumption of these books is that systems development is currently carried out using graphical tools that are drawn and maintained manually. The addition of computer support is a logical extension of practices already in place. This assumption has not been explicitly tested, as discussed in the earlier sections of this chapter.

Both Gane and McClure provide overviews of the "state of the art" in computer-aided software engineering (Gane, 1990; McClure, 1989). Stone (1993) takes a different view, focusing on two software vendors only: Texas Instruments and KnowledgeWare. These integrated computer-aided software engineering, or ICASE, products provide support through all phases of the life cycle. Stone compares and contrasts the way the two have implemented information engineering methodology.

Many researchers, writing for practitioners, have recommended approaches for adopting computer-aided software engineering technology. Recommendations for CASE adoption fall into four general categories: management commitment and support, technology infrastructure, training, and CASE implementation strategy.

Management commitment and support can take many forms. At the minimum, management must be willing to spend the time necessary to allow CASE adoption to "take hold." Some suggest that the adoption of CASE technology involves changing the culture of the organization, and that such cultural change will not come quickly (Brown, 1991; Orlikowski, 1993; Stone, 1993). In part, this change comes because the use of CASE tools requires a discipline that may not be present in the organization prior to the introduction of automated support. In addition to a willingness to wait for results, management must provide other resources, such as money for training (Yuen & Spurgeon, 1992).

The technology infrastructure is also important. Without a strong methodology, a uniform development process and consistently enforced standards, CASE adoption will not be successful (Rowe, 1993). Many organizations don't realize that they are not using a uniform development process until the adoption of CASE technology, at which point the

technology becomes the scapegoat for the lack of a unified systems development approach. The need to follow a methodology is more important with an ICASE tool, since the tool embeds the methodology within it (Brown, 1991).

Potential tools should be analyzed for their functional capabilities and for the fit between the capabilities and the requirements of the site (Bachman, 1988). Checklists of features and functions to expect in a CASE tool are found in the literature (Gibson, 1988; McGrath, 1993). These checklists were developed primarily using the researchers' personal experiences. These might be useful in providing a starting point, but researchers uniformly agree that there is no substitute for carefully determining what the organization expects to gain from the adoption of CASE technology (Gibson, Snyder, & Rainer, 1989; Norman, Corbitt, Butler, & McElroy, 1989; Rowe, 1993; Yuen & Spurgeon, 1992).

Both the technical staff and end users need to be trained in the use of the tools. Tools chosen without adequate support from both communities will not be accepted (Norman, et al., 1989; Orlikowski, 1989; Ryan & Bock, 1992). The lack of acceptance these tools receive will not be a function of the tools, but a reflection of the process by which the tools were introduced into the organization. The "right" tools chosen by the wrong people, or in an inappropriate way will not be used successfully. End users need to receive training in the way the tool will be used. This is especially true when end users will be expected to develop systems using CASE technology. They also need to be included when what they receive, either in terms of documentation or final products, will be affected by the technology.

Training is an important part of the CASE adoption process. Some researchers suggest that one to two weeks of training should be allotted per developer (Yuen & Spurgeon, 1992). Others suggest that however much training is allotted, the important factor is the timing. Training should be conducted as close to the actual implementation as possible (Brown, 1991). This is often called "just in time" training. While this advice is reasonable, practitioners aren't told how to determine the best possible time for training. Training delivery continues to be an issue since training is a large expense, and one that is often not adequately funded.

The CASE implementation strategy chosen also may play a part in the successful adoption and implementation of CASE technology. While many researchers argue that using CASE on a pilot project is the best way to implement the technology (Brown, 1991; Burkhard, 1989), others suggest that any strategy will work as long as there is a planned implementation, with follow up and measures of effectiveness (Yuen & Spurgeon, 1992). CASE implementation is different from project implementation, in that the application system might not be successfully implemented, but the implementation of CASE tool might succeed.

### Empirical Research

Empirical work consists of surveys, field studies and case studies. Surveys have been conducted to determine who is using CASE. One survey found that 24% of the respondents say they are using CASE tools (Necco, Tsai, & Holgeson, 1989). Those who are using CASE tools say that they have improved analysis and design, and improved communications and documentation (Necco, et al., 1989, p. 19).

A more recent survey of users indicated that CASE tools aren't being used to their full potential, since CASE is being used mostly for lower level

tasks, and "the more sophisticated CASE tools, such as code generators, are used substantially less" (Yellen, 1992, p. 39).

The most extensive field work has been done by Orlikowski, who has examined the interaction between CASE technology and the environment in which CASE is implemented (Orlikowski, 1989; Orlikowski, 1988; Orlikowski, 1993). This field work has included ethnographic studies and grounded theory work developing a theory of CASE as organizational change. The socio-technical aspects of CASE are complex. Roles change when CASE is introduced, and unless the impact of these changes is acknowledged, resistance by developers and end users can lead to CASE being adopted but not implemented, or not adopted.

A review of the case studies in the literature indicates that systems development is not carried out in a disciplined fashion, following a systems development life cycle, using a systems development methodology. These case studies contradict the earlier field research that assumed that firms were following an information systems development life cycle for application systems development. Those researchers reporting on uses of CASE technology report that the lack of a formal approach for information systems development is one of the biggest obstacles in the successful implementation of the technology (Chaney, 1992; Todd, Coleman, & Shimonek, 1991; Van De velde, 1992; Zagorsky, 1990). Since most ICASE tools are based on an information systems development life cycle approach, these case studies suggest that the implementers of ICASE tools will face even greater obstacles than the implementers of other types of CASE tools.

## Grounded Theory

### Overview

Information systems researchers have been encouraged to employ a number of different methods. The discipline is in danger of being dominated by one type of research methodology, some researchers argue, and we are the poorer for it (Benbasat, 1985; Benbasat, Goldstein, & Mead, 1987; Borland & Hirschheim, 1987; Gorry & Scott Morton, 1971). The use of both quantitative and qualitative methods in information systems research has been urged (Kaplan & Duchon, 1988). Qualitative methods that have been suggested for use in information systems research are case studies (Kaplan & Duchon, 1988; Lee, 1989; Marcus, 1983; Yin, 1984), action research (Wynekoop & Conger, 1990), ethnography (Orlikowski, 1988) and grounded theory (Calloway & Arlav, 1990; Orlikowski, 1993).

### Theoretical Research

Grounded theory is a rigorous approach to qualitative analysis, originally developed for use in sociology (Glaser & Strauss, 1967; Strauss & Corbin, 1990). While studying health care professionals at work, Glaser and Strauss became aware of the inadequacy of the current quantitative analysis methods. They were also aware of the criticisms of the qualitative methods that were being used. These two researchers adapted their field methods to add an element of rigor that they believed was missing in other qualitative analysis approaches. They developed a way to "discover" the underlying concepts that are present in the qualitative data that are collected.

The approach provides a systematic way of identifying underlying concepts or themes that are indicated by the data. The qualitative data

might be collected through participant observation, interviews, content analysis or other means.

One of the challenges in dealing with such data is conducting a thorough analysis. First, data are coded in a process called open coding. Data are classified according to the category they signify. Then, data classified in one category are examined for the underlying patterns that connect them. This process is called axial coding. Constant comparison takes place as new data are being analyzed and compared to data that have already been coded. Once data in a category are reassembled in axial coding, data are compared to each other to determine their similarities and their differences. Categories are then arranged hierarchically, so that core categories emerge from the data. The relationship between and among the core categories and their subcategories serves to explain the phenomenon under study (Glaser & Strauss, 1967; Strauss & Corbin, 1990; Turner, 1983). A more detailed description of the grounded theory approach is provided in Appendix A. The work of Glaser and Strauss has been extended by Strauss (1987) and Strauss and Corbin (1990).

### Empirical Research

Although grounded theory was originally used in sociology, organizational researchers have adopted this method as well. The technique is particularly well suited for organizational studies of any type (Martin & Turner, 1986). Grounded theory has been used in information systems research in recent years to study how managers evaluate their decision support systems (Toraskar, 1990), how designers use dialog charts (Calloway, 1988), how novice users interact with a hypertext help system (Fenton, 1992), and how CASE might be understood as a method of organizational change (Orlikowski, 1993).

Turner reports on a study of the classification of control systems operating within an organizational hierarchy of small batch production companies (Turner, 1983). The study, conducted by Reeves and Turner, identified nine organizational features, that together resulted in the understanding of a collective strategy that was adopted in batch production factories as a way of avoiding the complex scheduling problems that the foremen were faced with. Turner concludes that the only way the underlying patterns of social order could have been discovered was through the use of grounded theory to analyze the series of interviews.

Martin and Turner (1983) cite many studies that use grounded theory in organizational and professional settings. The studies that were cited span a period of about 20 years, establishing the method as one with longevity. The recent studies in information systems by Calloway and Fenton applied grounded theory techniques using students as subjects to determine how dialog charts were used (Calloway, 1988) and how novice learners used hypertext (Fenton, 1992). In both cases, field work was used to collect data that were analyzed using a grounded theory approach.

The more recent study by Orlikowski (1993) is much more ambitious. Orlikowski studied two different organizations, a software consulting firm and the information services division of a petroleum products company. The focus of this research was on the critical elements that shape the organizational changes that are associated with the adoption and use of CASE tools. Using grounded theory, Orlikowski articulated a theory of CASE as organizational change. Her work led her to conclude that the nature of the change will fall on a continuum between radical and incremental, and that either the process or the products of systems development will also change. Orlikowski states that the changes

surrounding CASE technology "emerge from particular interactions of institutional context, key players' intentions and actions, and the CASE technology" (Orlikowski, 1993, p. 333). Among the contributions of this work is the use of grounded theory in information systems research.

### Conclusions

While research has been conducted into the use of the systems development life cycle, methodology, and CASE, little empirical work supports or refutes the recommendations made by researchers. The available empirical research into CASE implementation is contradictory, and appears to be based on the assumption that developers are following earlier prescriptive advice to use the systems development life cycle approach and to follow a methodology. Case studies indicate that application systems are not developed following the discipline of methodology applied within the phases of an information systems development life cycle. This lack of a disciplined approach is an obstacle to successful implementation of CASE technology. Successful CASE technology implementation is the result of an interaction of factors. Grounded theory has been shown to be a technique of qualitative analysis that can be useful for organizational study, particularly in fields where there has been little theoretical development. Recent studies have demonstrated the effectiveness of this approach in information systems research.

The next chapter discusses the way that grounded theory was used in this study, including the method by which participants were identified and the kinds of analyses that were conducted.

## Chapter Three: Research Methodology and Design

### Introduction

This chapter discusses the research question, methodology and design, including how participants were identified and how the interviews were conducted. The next chapter discusses the analysis of the data, and the following chapter discusses the conclusions and suggestions for future research.

The relationships that were studied are identified and discussed. This chapter includes a discussion of the analytical procedures that were used to examine the relationships. The analysis of the interview data using the principles of grounded theory is described. The results of the analysis are presented in chapter four. A summary of the principles of grounded theory is presented in Appendix A.

### Research Question

**The purpose of this study is to explore the organizational and technological factors that contribute to the successful implementation of CASE technology.** The implementation of ICASE tools was examined through interviews with current and former ICASE tool users in organizations. The literature review suggested that the principal factors contributing to successful implementation fall into the following classes: management commitment and support, technology infrastructure, training, and CASE implementation strategy.

Implementation success is viewed as being dependent on factors from these groups. This research was conducted to explore the relationships between factors in each of these groups and implementation success. Questions in the interviews were included to explore these classes of factors. The kinds

of questions which were asked in each of these groups is explained in the next section of this chapter.

Management commitment and support is investigated by exploring the role of a champion. The designation of a champion is an indication of management support. The relationship between the presence of a champion and CASE implementation success was explored. Firms where a group or an individual has been designated as the champion of the new technology are theorized to be more likely to succeed (Burkhard, 1989; Lederer & Nath, 1991). Ideally, the CASE champion would be chosen from the more productive staff members. Others would follow the champion's lead. The position should also have resources and authority, which are evidence of management commitment.

Technology infrastructure is investigated by studying the role of an information systems development methodology. The relationship between adhering to a systems development methodology before the adoption of CASE technology and successful CASE technology implementation was explored. A finding of this study is that CASE supports existing good practice, but is not a substitute for good practice. Firms expecting CASE technology to provide both the underpinnings for good analysis and design and the discipline which is required for successful information systems implementation will be disappointed (Gane, 1990; Gibson, Snyder, & Rainer, 1989; McClure, 1989; Norman, Corbitt, Butler, & McElroy, 1989). A tool assists an organization in adhering to the discipline required to develop information systems. Some tools enforce the methodology more rigorously than others, but no tool is a substitute for sound software engineering practices of developing standards, adhering to a systems development life cycle and instituting quality assurance policies.

Training is studied by examining the mode and method of training delivery. Questions in this area relate to who conducted training. Is there any relationship between the form of training and implementation success? Training may be done in-house, by consultants, by the vendor, or in combination. Firms must prepare both the technical staff and end-users for the introduction of CASE technology. Management must be willing to have the first project take longer to complete than subsequent projects will take, while members of the team learn how to use the tool (Burkhard, 1989; Gibson, et al., 1989).

Implementation strategy is studied by examining the role of a pilot project in implementation. What is the relationship between pilot projects and CASE technology implementation success? The literature prescribes that CASE technology should be introduced using using a pilot project which can be carefully monitored and contained (Brown, 1991; Burkhard, 1989; Chen, Nunamaker, & Weber, 1989).

#### CASE Implementation Success

**The phenomenon studied in this research is ICASE implementation success.** Success can be difficult to measure. For consistency across organizations, a simple working definition of success that respondents would easily understand was required. Therefore, success was defined as the respondent's perception of the organization's level of satisfaction with its implementation of ICASE technology. The words "success" and "failure" are value-laden. A respondent may not want to report that the company "failed" but may be willing to report that the company was "very unsatisfied" with their implementation of ICASE technology. The question was framed so that the focus is on the implementation of the technology, and away from the organization: "In

general, how satisfied would you say your organization is with its implementation of ICASE technology?" The question was worded in this way to allow the respondent to comfortably report his or her candid assessment of the firm's overall situation.

#### Classes of Factors

The initial factors studied fall into four classes: management commitment and support, technology infrastructure, training, and CASE implementation strategy. These factors were suggested by a review of the literature, as discussed in chapter two. Yuen and Spurgeon (1992) in particular suggest that these four areas are among the most significant.

#### Management Commitment and Support

The existence of a champion was used as an indicator of management commitment and support. Respondents were asked directly if a champion was designated as a sponsor of the ICASE tool.

Providing a clear mandate for the use of the technology is a way of showing commitment. Choosing a champion whose mission is to ensure that the technology is implemented is evidence of both commitment and support. A champion is an advocate, an individual who has been designated to visibly promote the new technology. In this capacity, a champion is more than a leader. Often the power of a champion is a direct reflection of the respect he or she has earned in the organization. If the champion's position is high in the organization's hierarchy, the respect comes with this position. If the champion's position is lower in the hierarchy, he or she may have earned respect for being the "best" technically, or the most effective, or the most successful. A champion who doesn't have the respect of members of the organization is less likely to be successful.

Such a person also needs resources. Designating a champion without assigning resources (budgetary, personnel) defeats the purpose of assigning a champion. In addition to asking about the existence of a champion, participants were asked about the champion's position in the organization, the resources which were given to the champion, and the champion's role in CASE technology implementation.

### Technology Infrastructure

Respondents were asked about the use of a systems development methodology both before and after ICASE implementation. A finding of this research is that there is confusion among CASE users about the relationship between a CASE tool and a systems development methodology. Organizations that do not adhere to a systems development methodology will not enjoy the benefits of CASE technology implementation (Rowe, 1993). The discipline of systems analysis and design is supported by the tool, but not necessarily enforced by the tool. A CASE tool alone cannot be used to enforce an information systems development methodology. However, organizations that are adhering to an information systems development life cycle and a methodology will be more successful in their systems development activities with a CASE tool to assist them.

CASE users reported mixed use of an information systems development life cycle. Some users follow both a life cycle and an information systems development methodology, while other organizations use only an information systems development life cycle with no enforced methodology. Users whose use of a methodology was compatible with the level of methodology enforcement reported a high level of satisfaction with their implementation of CASE technology. Users who didn't use either a

life cycle approach, or an information systems development methodology found CASE tools too restrictive.

### Training

Respondents were asked about training for ICASE technology. This research investigated the relationship between who provides the training and the success of ICASE implementation.

The implementation of ICASE technology may dramatically change the way the organization develops information systems. Both technical staff and end users need to participate in some type of training. If a firm is already using a particular information systems development methodology, then the training is focused on the mechanics of using the particular tool. If the firm is not using a methodology, then training first must be conducted in the development of information systems following a methodology before tool training can take place.

Training can be handled in many ways. At issue are the extent and the quality of the training, as well as who provides the training. If management in an organization understands the implication of information technology use, training will be perceived as an important part of the implementation of ICASE technology. If management lacks understanding of information technology, formal training can be seen as a dispensable item and be replaced by "on the job training." There is general agreement that the vendor will provide the best training in the tool (Brown, 1991; Hayley & Lyman, 1990). However vendor training is typically the most expensive training. Consultants who understand both the tool and the underlying methodology are often contracted to conduct training. This is especially true if training is needed in both areas. Large organizations with training staffs sometimes conduct ICASE training in-house.

Participants reported various experiences with training. Some training was viewed by those who received it as inadequate, focusing on the mechanics of tool use when what was needed was a discussion of information systems development techniques. Many users reported struggling with the concept of delivering training "just in time."

### CASE Implementation Strategy

Respondents were asked directly about the implementation strategies used in their organizations. A review of the literature suggests that the use of a pilot project will usually lead to CASE implementation success (Brown, 1991; Hayley & Lyman, 1990; Rowe, 1993; Souza, 1990; Zagorsky, 1990). This research investigated the relationship between the implementation strategy chosen and the level of success.

Once the tool has been chosen, an implementation strategy must be selected. Here the focus is on implementing an ICASE tool. A pilot project is often chosen, and a full "roll out" of the technology is dependent upon the successful implementation of the technology for the pilot project. Other ways to implement ICASE technology include allowing the project team to choose the technology or mandating that all new projects will be developed using ICASE tools. A finding of this research is that a successful pilot project is not an indicator of implementation success. If the pilot project is not chosen to reflect the types of projects that will be conducted using CASE tools, the pilot might succeed but implementation of the tool will not.

## Research Methodology

### Data Collection

Data were collected through telephone and face-to-face interviews with current and past ICASE users singly or in groups of two.

Interviewing ICASE users provided an opportunity to explore questions in depth. Some interviews were tape recorded. Field notes were taken to record additional organizational information.

Twenty interviews were conducted with individuals from different types of organizations. Participants were chosen across a sample of organizations of different sizes, in both service and manufacturing industries. The data obtained from these personal interviews were analyzed using a grounded theory approach to develop a stronger, richer theory of how organizational and technological factors in organizations interact and contribute to successful implementation of ICASE technology.

To assist in identifying participants, a questionnaire was sent to members of CASE user groups. This questionnaire was used to develop a preliminary profile of CASE users. Interviews were conducted with users who were chosen to represent the typical CASE user as defined by this profile.

A search of the literature was conducted for questionnaires used to study CASE technology implementation. Previous researchers using questionnaires addressed the questions of who is using CASE tools (Necco, Tsai, & Holgeson, 1989) and what users think of the tools they are using (Yellen, 1992). Neither of these questionnaires examined the factors contributing to successful implementation of CASE technology. Therefore a new questionnaire was developed, based on a review of the literature and interviews with seven information systems professionals. The information systems professionals were chosen because of their experience with CASE technology. These interviews explored the factors these practitioners felt contributed to successful implementation of ICASE technology. They were then asked if they agreed with the list of factors that had been culled from

the literature. In this way the number of questions on the questionnaire was restricted to those that would provide relevant information. A question was eliminated if no more than one person felt it referred to a factor that contributes to successful implementation.

Four other people then reviewed and completed the pilot questionnaire. Of these four people, an information systems professional currently working with CASE tools and three colleagues of the researcher, only the information systems professional was interviewed and is included among the twenty interviewees. They assisted in questionnaire development by reviewing the pilot for clarity, layout and length of time to complete the questions. On the basis of their comments, the questionnaire was revised a final time. In these ways the questionnaire was tested to ensure that the correct factors were being investigated, and that the questions were clear, concise and could be answered in a fairly short time (which was about ten minutes). This final questionnaire is attached in Appendix B.

Descriptive statistics were used to develop a profile of CASE users from the questionnaire data. The relationship between each factor and CASE implementation success was examined using cross-tabulations. The respondent's opinion of the level of organizational satisfaction with ICASE implementation was used as the surrogate for success. The four classes of factors were derived from the literature, as discussed above: management commitment and support, technology infrastructure, training, and CASE implementation strategy. Data related to these factors were collected. The data were primarily ordinal along a Likert scale. The quantitative analysis was completed using SAS, a commercially available statistical analysis software package that contains a number of pre-programmed statistical

routines. The indicator of success was correlated with the existence of a designated champion, the use of a systems development methodology prior to the implementation of ICASE technology, the method of training, and the implementation strategy. This preliminary analysis provided a profile of typical CASE users. The analysis design is presented here. The profile which was developed is presented in chapter four. The details of the analysis are presented in Appendix C.

. Interview sites were selected to correspond with this profile. Twenty semi-structured interviews were conducted by telephone and in person. The seven interviews that were conducted initially, as part of the process of questionnaire development, were also used to develop the interviewing protocol used in the later part of the study. Thirteen interviews, with different participants, were subsequently conducted, ranging in length from 30 minutes to one hour. Of these thirteen, five were tape recorded and later transcribed by the researcher. These transcriptions provided the data from these interviews. The data from the remaining eight interviews were taken from field notes. The interview protocol is presented in Appendix D.

The interviews were conducted with participants who were either recommended to the researcher by another participant, who indicated on the questionnaire that they were willing to be interviewed, or who were chosen by the researcher from either a list of CASE users or members of a university advisory board to a school of Information Systems and Computer Science. Interview sites were chosen to present as wide a range of sites as possible, given the limitations of both time and resources. Respondents were chosen to represent a broad base of ICASE users as identified by the questionnaire profile. A general description of the

participants is included in chapter four. This "theoretical sampling," as it is called in grounded theory, allowed the collection of data from companies across industries and from companies that had been using the technology for both long and short periods of time.

Interviews provide a richness of data that is not possible when respondents fill out a questionnaire. Interviews are both time and labor intensive, and therefore fewer can be conducted, relative to the number of questionnaires that can be distributed. Techniques for analyzing the responses are fewer. However, interviewing allows for a rich, meaningful analysis of the reasons why some companies have successful experiences with ICASE technology while others have less successful ones.

#### Analysis Method

The interview data were analyzed using the principles of grounded theory. The grounded theory approach was first used in sociology in areas where theory development was weak. This method is appropriate in other situations for theory development as well, and has been widely used in organizational studies. The approach provides a rigorous, systematic way of analyzing interview data and categorizing the data empirically (see Appendix A). Through processes of open coding and axial coding, interview responses are analyzed to determine what core categories exist. Each interview is analyzed first to determine what conceptual categories the data fall into. This is the process of open coding. Then the data from each interview are grouped together to support the emerging categories in the process of axial coding. Using these categories, a theory can be developed about the phenomenon under study. The properties of the categories and the dimensions of the properties emerge from the collected data. The researcher integrates these categories, and their properties, in a

meaningful way to "tell a story" about the phenomenon under study. This "story" is the basis for the developing theory.

### Conclusions

This research used qualitative analysis techniques. A profile of ICASE users was developed from questionnaire data. This profile was used to select participants to interview for data collection. A grounded theory analysis was performed using the transcripts and notes of each interview. Qualitative methods are particularly appropriate for the development of theory. Little theory development has been done in the area of CASE implementation research. Grounded theory was chosen as the analysis method because of its rigor.

The next chapter discusses the data analysis and research results. The profile of CASE users that was developed is explained, and the way that this profile was used to identify interview participants is discussed. The analysis of the interview data is presented, along with the explanation for successful CASE implementation that was developed based on this analysis.

## Chapter Four: Data Analysis and Research Results

### Introduction

This chapter discusses the data analysis and the research results. The theory of CASE implementation that was developed based on the interview data is explained. This explanation describes the interaction of organizational and technological factors in the successful implementation of ICASE technology.

The CASE user profile that was used to identify both potential participants and questions to be discussed in the interviews is presented first. The questionnaire data collected in the categories of management commitment and support, technology infrastructure, training, and implementation strategy are discussed. The details of this data are presented in Appendix C. The use of grounded theory to develop the explanation is discussed next. Grounded theory is based on the discovery of categories from the data. This approach is a rigorous qualitative analysis method that has been used to study organizations (see for example Calloway and Arlav, (1990), Martin and Turner, (1986), Orlikowski (1993)). The principles of grounded theory are explained in Appendix A. The categories that emerged from the interview data and the major themes indicated by the categories are identified and discussed. Empirical support for the categories is illustrated using the interview data. The final section describes the theory that emerged from the data.

### CASE User Profile

This section discusses the CASE user profile which emerged from an analysis of the questionnaires. The questionnaire is included in Appendix B, followed by a discussion of the descriptive statistics in Appendix C. The purpose of the questionnaire analysis was to develop a general profile of

ICASE users, to determine with whom interviews should be conducted, and to define areas to be explored in these interviews.

An analysis of the data collected for the four classes of factors identified in the literature follows. The four classes are: management support and commitment, technology infrastructure, training, and implementation strategy. Figure 1 on page 79 shows the model suggested by the literature.

#### Management Support and Commitment

The designation of a champion served as an indicator of management commitment. A cross-tabulation of "champion" and "success" indicates no relationship between those respondents with a champion and the level of success that the respondents reported. Of those who reported being somewhat satisfied with their implementation of ICASE technology, two thirds had a champion and one third didn't. Although the small sample size precludes drawing any inferences about the relationship between the presence of a champion and success, a preliminary analysis of the data indicates that the presence of a champion is not a sufficient indicator of probable success.

However, since the survey also indicated that the role of a champion in CASE technology implementation is a complex one, this area was explored in depth in the interviews.

#### Technology Infrastructure

The frequency of use of an information systems development methodology prior to the implementation of ICASE technology was used as an indicator of the existence of a supporting infrastructure for the technology.

Ninety percent of the successful organizations reported that they "often" or "usually" used a systems development methodology prior to the implementation of ICASE technology. Thirty seven and one half percent of those who "never" used a methodology reported successful implementations.

The small sample precludes drawing any inferences as to the relationship between the use of a methodology before the implementation of ICASE technology and the level of success reported but it is apparent that there is likely to be a relationship under particular circumstances (Hayley & Lyman, 1990;. Rowe, 1993). Hence the use of a systems development methodology was explored in depth in the interviews. To account for both situations, interview participants were chosen from organizations that used an information systems development methodology and those who didn't.

### Training

Respondents were given a choice among the following alternatives for the delivery of training: vendor training, consultant training, "train the trainers," in-house training staff or other. These choices were not mutually exclusive.

No one method of training was preferred. Respondents reported using all the methods. Nineteen percent of the respondents reported that their organization used the vendor to provide training, and used no other training approach. Fifteen percent hired both the vendor and consultants, while another 15% used only consultants. "Other" was listed by 11.5% as the sole method of training. The remaining 39.5% were distributed among training the trainers as the only way training was delivered; using consultants and "other"; using the vendor and "other"; using the vendor and

an in-house training staff; using the vendor and training the trainers; using the vendor, training the trainers, and using the in-house training staff; using the vendor, using consultants and training the trainers; and one respondent whose organization used all five approaches.

The respondents who indicated that their organizations were successful reported using every type of training, alone or in combination. The wide distribution of training strategies suggests that the choice of strategy isn't as important as the content of the training. This is indicated by the comments made during the interviews. The issue of training is really an issue of focus: should the training focus on the use of the tool, or should the training focus on the use of the methodology, and use the tool to illustrate the methodology? The former is easier to accomplish. The latter takes place over time.

The information systems community has become more concerned with meeting deadlines and developing systems quickly. An information systems development methodology is seen as a hindrance, rather than a help, because of the time involved. Therefore, CASE training focuses on the use of the tool rather than the use of a methodology. The result is that most training appears to be of little use to those firms where the use of a methodology is an open issue. The CASE tool supports a systems development methodology, and where a methodology isn't being used CASE tool training isn't helpful. This may explain why so many respondents report using all types of training delivery methods. This analysis of the questionnaire data indicated that training was an issue to explore in depth in the interviews.

### Implementation Strategy

Although the literature recommends a pilot project as the approach for implementing CASE technology, respondents reported using a number of different strategies. Of the 38.5% who reported using a pilot project alone, half were successful. The same was true for the 15.4% who mandated CASE tool use for all new projects. Sixty percent of the 19.2% who left the decision up to the project team reported a successful implementation. All the respondents who reported using a combination of a pilot project and mandated use were successful, but they represented a small number of the total, at 11.5%. Seventy-five percent of those reporting "other" ways of implementing were successful. This group, which represents 15.4% of the total, reported the highest success rate. These results indicated that implementation strategy was an issue to pursue in depth in the interviews.

### The Use of Grounded Theory

Grounded theory was used to develop an explanation of the interaction of factors in the successful implementation of ICASE technology. The details of this approach are presented in Appendix A. This method was chosen because of the rigor it provides for the analysis of interview data and field notes.

Twenty interviews were conducted. Seven interviews were preliminary in nature; these interviews were used to develop and test the interview protocol and the questionnaire to be used to develop the CASE user profile. Each of the remaining thirteen interviews lasted an average of 50 minutes. Interviews were conducted with members of project teams, individual developers, personnel at large firms, personnel at small firms, firms in manufacturing and firms in service industries. The interview

protocol which was used is included in Appendix D. Table 1 on page 76 summarizes the background of those CASE users who were interviewed and the types of firms that they represent.

Interviewees were chosen to represent projects and organizations across as many dimensions as possible. This was done to insure that the data that were collected would represent as wide a range as possible. In grounded theory terminology, this is called "theoretical sampling." Theoretical sampling is analogous to statistical sampling. The researcher collects data from as many different sites as deemed relevant for the topic being studied. When no new dimensions of a category emerge from the data, "theoretical saturation" is reached. At the point of saturation, the researcher can be reasonably confident that the data collection has reached its end.

The sites chosen in this study, while similar, were chosen for their range. This research is developing a theory about ICASE implementation. Therefore, only ICASE sites were studied. Other types of CASE sites were eliminated from consideration. A detailed discussion of theoretical sampling and theoretical saturation is provided in Appendix A.

#### Category Development from the Data

The objective of the first analysis step is to develop a set of categories to which the data belong. To do this, each of the interviews was analyzed using the process of "open coding." For those interviews for which transcripts were prepared, each sentence of the transcript was reviewed, and the comments placed in categories. Comments are "coded" as belonging to a category. A list of categories culled from the literature was used to start the process. The interview data revealed the existence of additional categories. Not all comments or notes belonged to a category.

Some comments were discarded as they were not relevant to this study. A set of code notes was prepared linking each interview comment to its appropriate category. A sample of code notes is presented in Appendix E. For those interviews for which a set of notes was prepared, the notes were analyzed in a similar manner. Field notes, which included the interviewer's comments and impressions, were also used to help determine categories. The process of open coding is explained in more detail in Appendix A.

### Categories

Table 2 on page 77 lists the categories that were developed from the data. These categories, which represent the underlying abstract concepts, are presented alphabetically along with their definitions. Some of these categories were derived initially from the literature while others were discovered in the data. Those in italics were discovered in the data.

The categories of champion, methodology, management commitment, pilot project and training were used to provide a starting point for categorizing the comments collected during each interview. The remaining categories emerged in the analysis of the interview data, as the interviews yielded some data that did not fall into the original categories. The comments collected in the interviews indicated that there were categories not found in the literature.

Four new categories were identified in the process of open coding. These are shown in italics in Table 2. The emergence of additional categories is expected in an area where limited empirical and theoretical work has been done. Grounded theory can be a useful methodological approach in helping to develop theories in these areas. This method was chosen for that reason. One of the original categories, pilot project, was

subsumed in the category of implementation. This process resulted in the categories listed alphabetically in Table 2.

Following the development of these categories, the second step, "axial coding," was conducted. The objective is to integrate and connect categories. In open coding the data are divided; in axial coding the data are re-united, in different ways, to uncover a pattern of integration among and between categories. The processes of open coding and axial coding lead to the last step of "selective coding" where the core themes are identified, and the categories are related to the core themes. This fairly complex step produces the theory based on the data collected in the interviews.

The categories were analyzed for their similarities to each other, as well as their differences. The categories were divided into groups of three. Each group, or triplet, was analyzed and the two categories that were most similar were identified. A triplet was also studied to determine which category was most different. This was conducted iteratively. In this way each category was compared with all other categories.<sup>2</sup> This analysis of the categories led to the discovery of three major themes: organizational understanding of information systems technology, information systems development environment, and information systems development complexity.

The section that follows discusses these core themes and provides examples of the empirical support for them. Samples of the supporting data are included in Appendix E.

---

<sup>2</sup>I am grateful to Dr. Linda Jo Calloway for suggesting this method.

## Core Themes

This research found that these three core themes are related to each other through a shared category, "methodology." Additionally, the core themes of organizational understanding of information technology and information systems development complexity share the category "benefits," and information systems development complexity and information systems development environment share the category "training." These shared categories relate the core themes. Figure 2 on page 80 illustrates the grouping of the categories into these core themes. Table 3 on page 78 illustrates the dimensions of the core themes.

### Organizational Understanding of Information Systems Technology

Organizational understanding refers to how well management understands the impact of CASE technology on the organization's information systems development activities. This includes all aspects of CASE adoption, implementation, and use. This research found that the categories that indicate organizational understanding are champion, decision factors, benefits, management commitment, and methodology. The methodology category relates organizational understanding to information systems development environment and information systems development complexity, and the benefits category relates it to information systems development complexity.

#### Champion

"Champion" refers to the presence or absence of an officially designated champion or sponsor for CASE technology. While the user profile indicates that 70% of all respondents had a champion, the qualitative data indicate that the presence of a champion isn't enough to ensure success. The champion must be at the right level in the

organization, with a certain amount of visibility and power. An analysis of the interview data relating to the presence or the absence of an ICASE champion reveals that the champion must have specific qualities that will allow the organization to be successful. A champion must be high enough in the organization to have power. As one respondent put it, "It really goes back to is there a champion at the top. If there is a champion and that person will make things happen pretty much regardless of the cost. If there's no champion then okay here is another expense."

The visibility of the champion is also important. Another respondent, in discussing why a project wasn't as successful as he thought it should have been, said, "The end user of this system wasn't someone who had the visibility to really champion it." One person who wanted to be the champion, believing strongly in CASE but unable to get his organization to "buy into" the use of the technology, summed it up by saying, "I was never in a position of high enough level to be a champion."

The choice of who will champion the technology is clearly more important than merely having a champion. Unless the champion is chosen wisely, the presence or absence of a champion will have no effect on the outcome of the implementation.

It is not necessary that the champion be a member of the organization. An outside consultant's preference, coupled with a strong reputation, can provide the power and visibility for the consultant to serve as the champion. Speaking about his firm's decision to move toward a new CASE product, one interviewee commented, "But what's championed that is we have an outside consultant who's coming in and doing some work and I guess he's familiar with [the product] and feels that's what he wants to do." In this case, the consultant is the champion.

The role of a champion might be taken by an individual or a group. In discussing his firm's decision to implement an information engineering based ICASE tool, one interviewee described the evaluation stage of the technology and the role of the champion by saying, "I don't think there was much emphasis on trying to champion at that stage, it was more on proving the technology and setting up the infrastructure." The pilot project stood in place of a champion. He noted that "a lot of people were looking at what we were developing, and if we were successful, maybe they would have come on board a little more." If the pilot had worked, everyone would want to use the technology. In this case, the failure of the project had little to do with the technology. In the words of the interviewee, "A lot of our issues weren't really CASE related. Some of the failure is not as a result of CASE or the [product]. Just difficulties in what we attempted to achieve." A finding of this research is that the presence of a champion isn't sufficient for successful implementation of CASE technology.

#### Decision factors

"Decision factors" refer to those elements that were considered significant in making the decision to adopt a particular tool. In some organizations, the underlying methodology used by the CASE tool is important. In others, the promise of a shorter information systems development life cycle, or higher quality systems, or increased programmer productivity lead management toward making the decision to purchase a CASE tool. Often the decision is made for what information systems professionals in the organization feel are the "wrong" reasons. In discussing the cost of the tool, one interviewee commented that "It [cost] wasn't a factor in the evaluation. Justifying the expense was the next phase." He went on to say later in the interview that "it [the purchase of

CASE technology] had already been prejustified before we went through the process, based on the wrong set of criteria: removes coding, removes maintenance." His organization was looking for maintenance free program source code that could be generated by defining business rules, somehow. He described this as, "They [management] also had this concept of just defining the business rules, press a button, and there was no coding."

This research found that if the organizational objectives are stated before the adoption of CASE technology, and the decision factors are measurable, then organization can determine whether they have achieved their objectives. Decision factors used "get lost" in the process of implementing a particular tool. For example, organizations choosing CASE tools to automatically generate program source code can easily tell if the objective is met: source code is generated automatically or it isn't.

Organizations choosing CASE tools to improve productivity can only determine if productivity is improved if they have some way to measure productivity before the use of CASE and they can determine that the increase in productivity is attributable to the use of the CASE tool. One interviewee stressed "I can't overemphasize the practical nature. Balancing the long term benefits with the short term benefits. How do you justify such a huge investment, slowing down a project? That's why we're looking for more incremental approaches rather than all or nothing." Later this same interviewee, when asked about the specific reasons that the organization ranked as important in determining whether or not CASE would be used, said "I don't think that we have approached the -- specifically isolated what CASE would do in those areas. We haven't focused on what CASE is really doing for us."

## Benefits

The benefits that management expects to achieve can also be crucial in the success of CASE implementation. Managing those expectations is a difficult, and important, job. The interviewee from the organization where the management team felt that CASE was going to allow them to just "push a button" summarized with an understatement, saying, "In terms of their original expectations, they were a bit disappointed." Often expected benefits are that programmers will not be needed, that a lower level of skill will be required among information systems professionals working in the organization, and that the information systems development life cycle will be shorter. Some of these benefits may be achieved in the long term, but not on the first project, nor even perhaps in the first year. Some of these perceived benefits will probably never be achieved. Information systems will eventually be developed more quickly, due to the development of repository data that can be used in future projects. However, programmers are often still needed to maintain even generated source code, and the skill level required to efficiently use complex CASE tools can be higher than the current skill level of the information systems professionals in the organization.

Some information systems professionals believe that management is not as willing to wait for results now as they were in the past. CASE tools provide a way to benefit not only the current project but future projects. The temptation to do the "best" job to meet both short term and long term objectives may be hard to resist. As one interviewee phrased it, "one of the difficult challenges in each organization ... that's knowing when to move out of analysis, you know, I'm sure you've heard of analysis paralysis. I think that's even more true in CASE. The initial team is almost

brainwashed into this vision that they have to strive for 100% quality, reusability, managed data environment. They believe that CASE can help them reach that vision. Management comes along and says no, we need the deliverable in six months or nine months and I think that there's an enormous conflict there." The conflict between meeting today's deadline and making the investment in CASE tools pay off becomes a choice between using the tool's capabilities for this project, or using the tool's capabilities for reusable modules of code, or design elements, or quality improvement techniques.

This desire for immediate improvement is exacerbated if the organization's information systems portfolio consists of complex application systems. Complexity is a function of the number of people on the team as well as the number of users affected by the system and the overall size of the project, whether measured in terms of lines of code or a more sophisticated metric. In an organization where simple systems are developed, some of the "up front" benefits can be realized much sooner. Project team members may not require extensive training to use the tool, since they will not be making use of all the tool's features. A single developer using a CASE tool can develop an application more quickly with the tool than without it. In the words of an interviewee, discussing a time when he rushed a design instead of using a tool, "If I would have done the proper design up front, instead of taking one day to do it, I would have taken the three days I should have done. Instead of programming for eight days I would have programmed for three days." This shortened time frame can be substantial if the tool generates source code that, prior to adoption of the tool, the developer would have written and tested.

### Management commitment

"Management commitment" refers to the ability of the organization's management team to ensure that the correct resources will be allotted to the project and that the proper amount of time will be spent on the activity. The investment in CASE technology requires both a long-term focus and some organizational change. One interviewee was very clear that unless management commitment was expressed in terms of incentives, commitment wasn't there: "If they were to turn around and say, not only are you incented [sic] on making your deadline, you're going to be incented on how much reusability you've achieved from looking at the corporate models that are in existence and reusing the existing models, the existing routines out of a library, whatever it is that took place. That's got to be a part of it." In his organization, the long term benefits are being given up to meet short term objectives.

### Methodology

Conflict can occur when management doesn't really understand what CASE can and can not do for the organization. The use of an information systems development methodology to develop application systems can assist in both project management and organizational understanding. If a methodology is adhered to, and the deliverables of the methodology are standardized and considered an expected piece of the application system, the organization as a whole has a better understanding of the information systems development process. If a methodology is not adhered to, then members of the organization may have little or no understanding of the steps involved in developing and implementing information systems.

### Information Systems Development Environment

Information systems development environment describes the context in which information systems are developed. Each organization has its own environment that affects the ways in which all information systems activities are carried out. This research found that the categories that describe information systems environment are implementation, methodology, skill set, and tool use. Methodology relates this theme to organizational understanding of information technology and information systems development complexity, and implementation strategy relates it to information systems development complexity.

#### Implementation

The implementation strategy used is an indicator of the information systems development environment. Firms can choose to implement ICASE technology by choosing a pilot project, by mandating its use, by letting a project team decide to use the tool, or by some combination of these methods. Many organizations choose pilot projects. A pilot project may not be suitable for all organizations. As one interviewee remembered, "Initially people wanted to join the project. It was using DB2, it was using CASE, people thought this would be really good and then I think as some of the earlier mistakes we made and people could see it wasn't really moving as fast as everybody wanted it to move, then I guess people started distancing themselves from the project. People thought you didn't need to know SQL, you didn't even need to understand relational DBMSs, [all] you had to know how to produce a data model. People who had been producing data models were very theoretical." In this particular organization, the respondent felt that the wrong project was selected for the pilot. He described it as "the wrong project with the wrong time-frames;

we just picked a date before we knew what our focus was." Other divisions of the organization saw a big investment with no benefit, and "they were a little put off, but they had nothing better to offer, either." This division of the organization was committed to using Information Engineering as a methodology, spent time conducting a thorough evaluation including building a small project in each of three ICASE tools, and then chose a pilot project that was too ambitious given the visibility and the pressure. The pilot was never completed, and the project focus was changed from developing an application to purchasing a package.

The pilot project's failure, in this instance, had more to do with the specific application than with the use of a CASE tool. The application system under development might be a failure but the implementation of CASE technology could still succeed. The separation of these two events is sometimes clouded by the outcome of the application system development project. In the situation cited earlier, the project failure left some of the organization's information systems professionals unwilling to try the tool because they associated the tool with the project. Since the project failed, the tool must be a failure.

This research found that more is involved than the method of implementation. The pilot project must be analyzed to determine why it was successful, and how well the characteristics of the pilot match the development project portfolio of the organization. Interviewees stressed the importance of choosing the correct pilot project, learning from the pilot, and applying the knowledge to later projects. Often this is not done, according to the data collected in the interviews. As one respondent phrased it, "We tend to say it worked for the pilot, it will work everywhere. That's not the case." Not only is that not the case, but this

person went on to say, "Even where you have a success, we don't go back to see why we were successful."

This research also found that organizations should plan how they will use the tool once the pilot has ended. One reason that a pilot project succeeds but implementation doesn't is that the pilot is carefully chosen and monitored. As one respondent pointed out, "pilot recommendation: start small, isolated, stand alone. Found companies chose very small, very low profile. Have a hard time moving into the next project. Tool hasn't got the profile required to make changes internally. Experience in small environment doesn't provide a stepping stone in between large project." The idea that a small, isolated, stand-alone project could serve as a stepping stone to larger projects doesn't hold up, but a more complex pilot project is much harder to monitor.

#### Methodology

Another indicator of the environment is the degree to which a methodology is followed. Some organizations follow a particular methodology, carefully enforcing all the activities prescribed by the methodology. Other organizations follow no methodology at all. Still others fall somewhere in the middle. A methodology might be followed somewhat, but not rigorously, not all the time, and not on every project.

A finding of this research is that the information systems community has confused methodology and CASE tools. This issue arose in many of the interviews. The underlying assumption of CASE vendors is that users are employing an information systems development methodology. In most situations, the ICASE market relies on Information Engineering, as developed by Martin. This research suggests that assumption is false. This

assumption explains what many people see is "wrong" with ICASE tools as are they are currently being marketed.

The issue of information systems development methodology is particularly important when ICASE tools are being used because ICASE tools are closely coupled with a methodology. Some tools enforce the methodology rigorously. In an organization where a methodology is not followed, or followed only some of the time, these tools will appear too rigid. In an organization used to the discipline of a methodology, these tools will be seen as assisting in the enforcement of the methodology, and will be viewed as helpful.

#### Skill set

The skill set of the information systems personnel is another indicator of information systems development environment. The skill set describes the set of skills that information systems personnel currently have and are expected to have in the future. In these organizations, the skill set consists of strong analytic and design skills. Nearly all the organizations that participated in this study indicated that information systems professionals were expected to accommodate changing technical environments by exhibiting a willingness to learn new skills. As one interviewee phrased it, "What do you need to be successful? Willingness to change. The people who realize that it isn't as different as you may think [that it is]."

#### Tool use

Finally, "tool use" describes how the information systems professionals in the organization plan to use the tool, as well as how they actually use the tool. CASE tools may also be used for automatic documentation, to model processes, to model data, and to maintain a data

dictionary. Many organizations use CASE tools for data modeling only. One interviewee suggested that this was preferable because data modeling was the most portable and least platform dependent information systems development activity. A few interviewees said that the primary use for ICASE tools in their organizations was the documentation of development deliverables.

Automatic program source code generation is one aspect of tool use. Code generation refers to the specific use of the tool to generate program source code, usually in a language such as COBOL. Some organizations have no programmers on staff. In these organizations, automatic program source code generation is a benefit, providing skills that the organization doesn't have. In other organizations, a division of labor exists that separates analysts, designers and programmers. An organization might pride itself on the quality of the source code written by its programmers. Such an organization will have little use for the code generation abilities of an ICASE tool. If the task of programming was to be eliminated, the skill set of that group of information systems professionals would have to change. Understandably, programmers often feel threatened when the use of ICASE tools for automatic program source code generation is expected. One respondent said that he "won't use the code generator." It is his belief that the organization has lost some good programmers merely by suggesting that automatic program source code generation was being considered. In his opinion, "code generation is too far for the tool to go." His organization has very few "orphaned" programs. Nearly all the programs that are being used in production are being maintained by the programmers who wrote them. These programmers can't (or won't)

change automatically generated code, which makes maintenance of the code difficult.

At the other end of the spectrum, in an organization with no programmers, code generation is viewed as an essential piece of the tool. In another company, with no programmers, the code was generated automatically. If the code needed to be changed, rather than changing the generated source code, the analysts changed the diagrams and specified that the code be generated again. To maintain program source code without the CASE tool this firm would have had to hire programmers. The expense was not worth the benefit, since programming was not a full time activity in the organization.

Some organizations fall between the "no programmer" position and the "programmer ownership" position. One respondent philosophically summarized his organization's position by saying, "Why do this [CASE] if you are not going to generate code?" As president of a user group for a large integrated CASE tool put it, "There are a couple of companies, if it's a very large batch process they use [the tool] to generate the code and then they go back into to fine tune and make some adjustments." Code generation is not a benefit for every firm. As one interviewee said, "you have to be a special organization to make it work, and that is not [his organization]."

The issue of tool use arose in a number of interviews as an important factor in ICASE implementation success or failure. Tool use was not addressed in the literature. Some organizations determined before the implementation of ICASE technology how the tool would be used. One respondent explained, as an answer to how the tool was implemented, that the tool was used "mainly for data modeling on new projects." The use of

a CASE tool for data modeling alone was mentioned in many interviews. Implementation strategies might be dependent on the expected use of the tool. Tools that are going to be used for data modeling portions of projects only, even though they can be used across the life cycle, might not need the same type of implementation approach that a tool that is going to be used to generate code needs.

### Information Systems Development Complexity

Information systems development complexity is the degree of complexity of the systems that the organization develops using CASE technology. This research found that the categories indicative of information systems development complexity are benefits, implementation, methodology and training. The benefits category is also related to organizational understanding; the implementation category is also related to information systems development environment. "Methodology" binds information systems development complexity to information systems development environment and organizational understanding of information technology.

#### **Benefits**

As discussed earlier, the benefits that an organization achieves depend in part on the complexity of the organization's information systems. The more complex the information systems an organization is developing, the greater the benefits the organization will enjoy. The development of a simple system that is expected to have a short life and no maintenance might be developed more quickly without CASE technology, but it can still be useful as a means of learning the tool, which can then be put to more effective use on larger projects..

The perceived benefits are affected by management expectations of what might be accomplished with CASE tools. Under some circumstances, any benefits achieved will be overshadowed by unrealistic expectations of the benefits that should have been achieved. Sometimes the expectation is that these tools are easy to learn, easy to use, and make the development of information systems a simple activity. The tools might be easy to learn and use, but the tasks at hand are complex. These expectations might be unrealistic, but management in organizations sometimes looks for a "knight in shining armor," or a "silver bullet." CASE technology is sometimes seen as the silver bullet. One respondent described the gap between reality and expectations at his firm, saying, "Even during the evaluation process, we spoke to a number of clients in the US and the UK just trying to find out their experiences, and I think full life cycle CASE was still relatively new in lots of those organizations, but they identified a number of the problems we came across, in terms of types of projects, difficulties of changing the environment. They were all identified to management. But management had this vision that here was CASE, it was going to save the day."

#### Implementation

Another aspect of complexity relates to the implementation method chosen for CASE technology. An organization might chose to implement CASE technology using a simple project as a pilot project. When the pilot project is successful, CASE might be "rolled out" to all other projects in the organization. If those projects are also "simple" projects, then the implementation will be successful. However, if the projects that are developed after the pilot project are more complex, the implementation throughout the organization will not achieve success. Complexity refers to

both the way in which the project team works and the output of the project team's efforts. Project team size can be a factor. Project team organization ranges from simple, one-person project teams to many-person development teams. Often a small project is chosen for the pilot, when the tool will be used to develop larger systems once the pilot has been completed. The pilot isn't analyzed to determine if the implementation was successful due to some characteristics of the pilot project that are not typical of the kinds of development projects for which the tool will be used. Organizations that chose to implement CASE technology by mandate might find resistance to such a mandate. A mandate undertaken without recognition of the complexity of the information systems development activity will not allow an organization to be successful at implementing CASE technology.

A finding of this research is that organizations should use pilot projects to implement CASE technology but these projects should be analyzed to determine their similarity to future projects. Where large differences exist, additional pilot projects should be considered.

### Methodology

The underlying methodology upon which CASE technology is based provides a deep understanding of the problem. A simple, single user system with a short expected life does not need the depth of understanding required for a more complex, multi-user system. One interviewee, in speaking about data modeling and diagramming with CASE tools, said, "I find number one it helps me understand it [the problem]. I find I believe in it. You know, I've seen enough bad things developed and I see what the benefits of data modeling are and I honestly use that to communicate not only amongst ourselves but also with end users." The methodology

provides both a tool for communication and increased understanding. These are related to the complexity of the problem being examined. The more complex the problem, the more understanding is needed.

Another respondent was more graphic. In speaking about using a tool without understanding the methodology he said, "It's like getting in a car and driving before you've had driving lessons....Eventually you're going to destroy it. But you know, you learn along the way. If you survive, you're a good driver."

### Training

Training and support requirements are a function of information systems complexity. Many organizations attempt just-in-time training, where information systems professionals receive the training they need just before they need it. Management of this activity is increasingly difficult in a complex environment, especially as resources become scarce. As one interviewee put it, "Overall in general, nobody has enough money for training. Or time. We're not good at just-in-time training. Our process leader group is working with our training group to determine when is the right time for training." The issue of money for training is becoming increasingly important.

The desire to save money has led some organizations to cut back on training, based on the idea that the tools are supposedly easy to use. "What they try and do is get people, either from outside or inside the company, to become trainers in [the training division] so that they don't have to hire these companies and pay the exorbitant rate that they claim these companies pay. What that ends up translating into, though, is a diminished curriculum with a diminished quality of training, more of a numbers game of how many people can we get through a set of courses at such and such a cost,

rather than how effective was the education that we put these people through." This same respondent, in speaking about management commitment to CASE technology remarked, "I can throw lots of bodies and say, why weren't you successful. You told me this case stuff was easy, anybody could do it. Isn't one of the benefits supposed to be that you don't need that level of a person, with that much experience?"

Another interviewee remembered that "The team that was set up to work on that pilot, they had two people who were previously used for training, probably Lotus spreadsheets, the thought was that these people would learn the tools and then train people on the tools. I think that quickly disappeared in that the technology is so much more complicated." As one respondent put it, "There's a gray line between teaching people how to use a tool and how to use a methodology...as a consequence you can draw the diagram but you don't understand the reasons and benefits."

The more complex the system, the more aspects of CASE technology that will be called into play. For some CASE tools, this requires that more actual coding must be done. This coding might be as extensive as writing source code for programs, and might involve the same skill level. In the words of one interviewee, "I think that there is also a feeling that the amount of code that you have to develop is probably a little more than what you want to be doing now. For example, new tools like Visual Basic have a lot more behind the scenes." For a more complex system, more training will be required.

Support requirements raise similar issues. One respondent said that he chose to use a particular CASE tool simply because "...I had a certain amount of support, somebody I could pick up the phone and call. And really, it just made sense." This respondent's level of comfort with the

support he knew he would receive allowed him to confidently proceed with the CASE tool, knowing that he would be able to get the help he needed if he had problems.

### A Theory of CASE Implementation

The interview data indicate that CASE implementation success or failure is the result of a complex interaction between the information systems development environment, the information systems development complexity, and the organizational understanding of information systems technology. This general framework partially supports the work of Orlikowski (1993) who found that to understand and explain the impact of CASE tools on an organization the social context of information systems development, the intentions and actions of key players, and the implementation process followed must all be considered. Orlikowski's research focuses on how CASE tools change information systems development. This research focuses specifically on successful outcomes. However, in both cases the organizational context and the implementation process must be considered.

The framework indicated by this data is presented in Figure 2 on page 80, which illustrates the categories making up each core theme. The boundary of each theme is represented by an oval. The overlapping ovals show how each theme is related to each other theme. The category "methodology" is contained within each of the three ovals, since this is the category which relates all three themes. The category "benefits" is shown in the oval bounding information systems development complexity and organizational understanding of information technology, since that category relates these two themes. The category "implementation" is shown within the boundaries of both information systems development

environment and information systems development complexity, since that category relates these two themes.

Each of these three core themes also has a set of dimensions, presented in Table 3 on page 78. The dimensions of organizational understanding run along a continuum from little understanding to complete understanding. Information systems environment runs along a continuum from disciplined to undisciplined, and information systems development complexity runs along a continuum from simple to complex. The end points of these continua are theoretical. For example, no information systems environment is completely disciplined nor completely undisciplined. These points represent the theoretical bounds of each core theme.

The top part of Table 3 indicates that a simple system, developed in an undisciplined information systems environment may be successfully developed regardless of the level of organizational understanding. A complex system, developed in a disciplined information systems environment, with a complete level of organizational understanding, has a high chance of success, while a complex system developed in an undisciplined information systems development environment with a low level of organizational understanding has a low chance of success.

The middle portion of Table 3 indicates that a simple system developed in an undisciplined information systems development environment has a medium chance of success, while a simple system developed in a disciplined information systems development environment has a high chance of success. Similarly, a complex information system developed in an undisciplined information systems development environment has a low chance of success, while a complex system

developed in a disciplined information systems environment has a high chance of success.

Finally, the bottom portion of Table 3 indicates that an information system developed in an undisciplined information systems development environment has a low chance of success regardless of the level of organizational understanding of information systems technology, while systems developed in disciplined information systems environment have a medium chance of success where there is little organizational understanding of information systems technology, and a high chance of success where there is a complete understanding.

The interaction between and among the factors indicates that an organization attempting to successfully implement ICASE technology must be aware of the role that each of these factors plays. In the use of ICASE, a methodology plays a critical role. Because of the integration among the phases, an organization that doesn't follow the underlying methodology of the tool will not be successful.

Following a methodology is a necessary but not sufficient condition of success. One consultant put it this way: "if the methodology doesn't support the application development paradigm upon which the tool is based...successful use of CASE will generally not be achievable" (Stone, 1993, p. 14). This research supports that statement. However, the organization must understand the implications of adhering to a methodology. The emphasis must shift from productivity to quality, from meeting today's deadlines to reusing design elements. The goal must be better quality information systems applications.

The final chapter of this dissertation summarizes these conclusions. The implications of this research for both practitioners and information

systems researchers is discussed. Finally, suggestions for future research are presented.

Table 1  
Background of Interview Participants

Position of Interviewee	CASE Tool Support	Applications Development	Total
Consultant		2	2
Senior Manager	4	3	7
Project Manager	5		5
Senior Analyst	6		6
Total	15	5	20

Number of Interviews by Type of Industry

<i>Service</i>	<i>Manufacturing</i>
14	6

Number of Interviews by Type of Firm

<i>For Profit</i>	<i>Not for Profit</i>
20	0
<i>Public</i>	<i>Private</i>
5	15

Years of CASE Use: Range 4-10

Table 2  
Categories and Their Definitions

The categories discovered in the data are shown in italics.

***Benefits:*** The benefits the organization expected to receive as a result of CASE technology implementation.

***Champion:*** The presence and role of a CASE champion

***Decision factors:*** The factors that were considered when the organization purchased the tool.

***Implementation:*** The implementation strategy for CASE technology used by the organization.

***Management commitment:*** Management's commitment to the implementation of CASE.

***Methodology:*** The role and importance of an information systems development methodology in the organization.

***Skill set:*** The set of skills which information systems professionals using the tools are expected to have.

***Tool use:*** The way the CASE tool is used in the organization.

***Training:*** The way training is conducted in the organization.

**Table 3**  
**Relationship Between Core Themes and Chance of Success**

<b>Information Systems Development Complexity</b>		<b>Organizational Understanding of Information Systems Technology</b>	
		<i>Little</i>	<i>Complete</i>
<i>Simple</i>		Medium chance of success	High chance of success
<i>Complex</i>		Low chance of success	High chance of success

<b>Information Systems Development Complexity</b>		<b>Information Systems Development Environment</b>	
		<i>Undisciplined</i>	<i>Disciplined</i>
<i>Simple</i>		Medium chance of success	High chance of success
<i>Complex</i>		Low chance of success	High chance of success

<b>Information Systems Development Environment</b>		<b>Organizational Understanding of Information Systems Technology</b>	
		<i>Little</i>	<i>Complete</i>
<i>Undisciplined</i>		Low chance of success	Low chance of success
<i>Disciplined</i>		Medium chance of success	High chance of success

The theoretical end points on a continuum are shown for each theme.

Figure 1  
Classes of Factors Suggested by the Literature

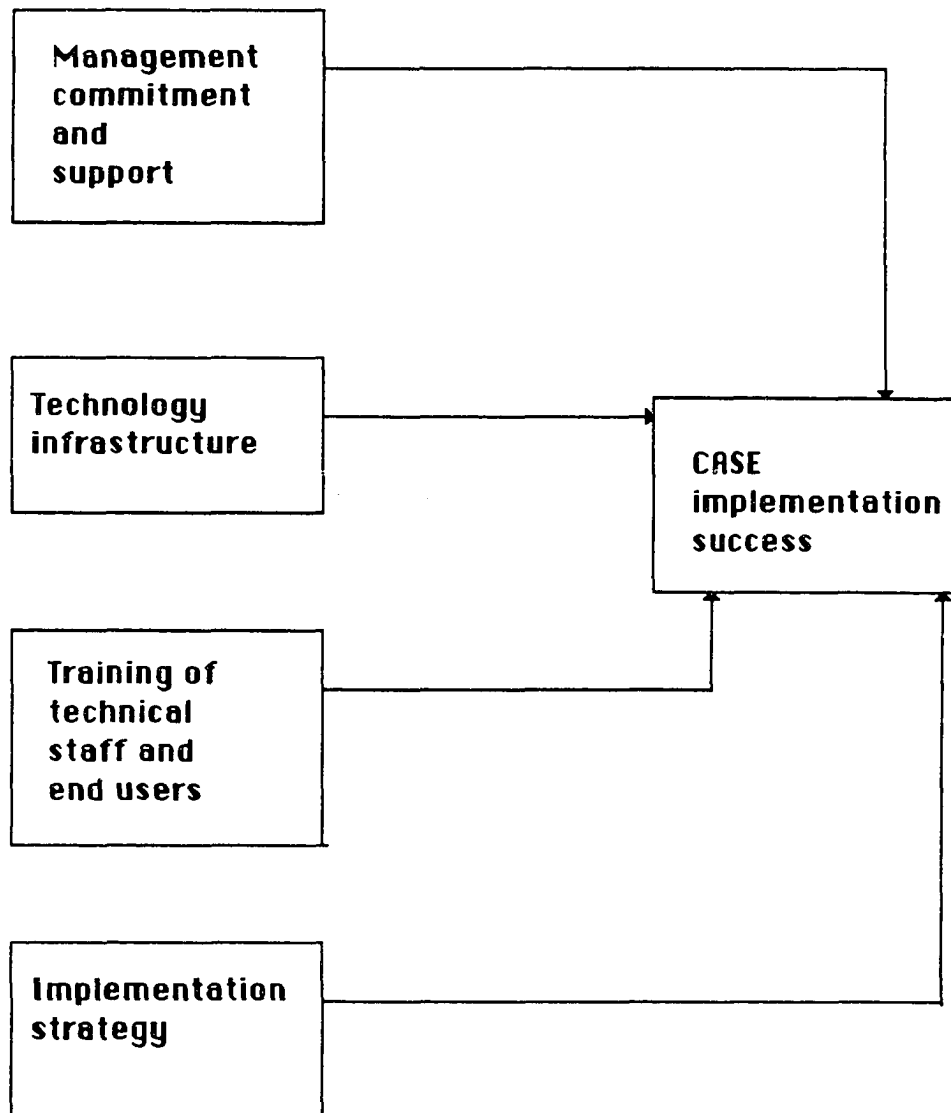
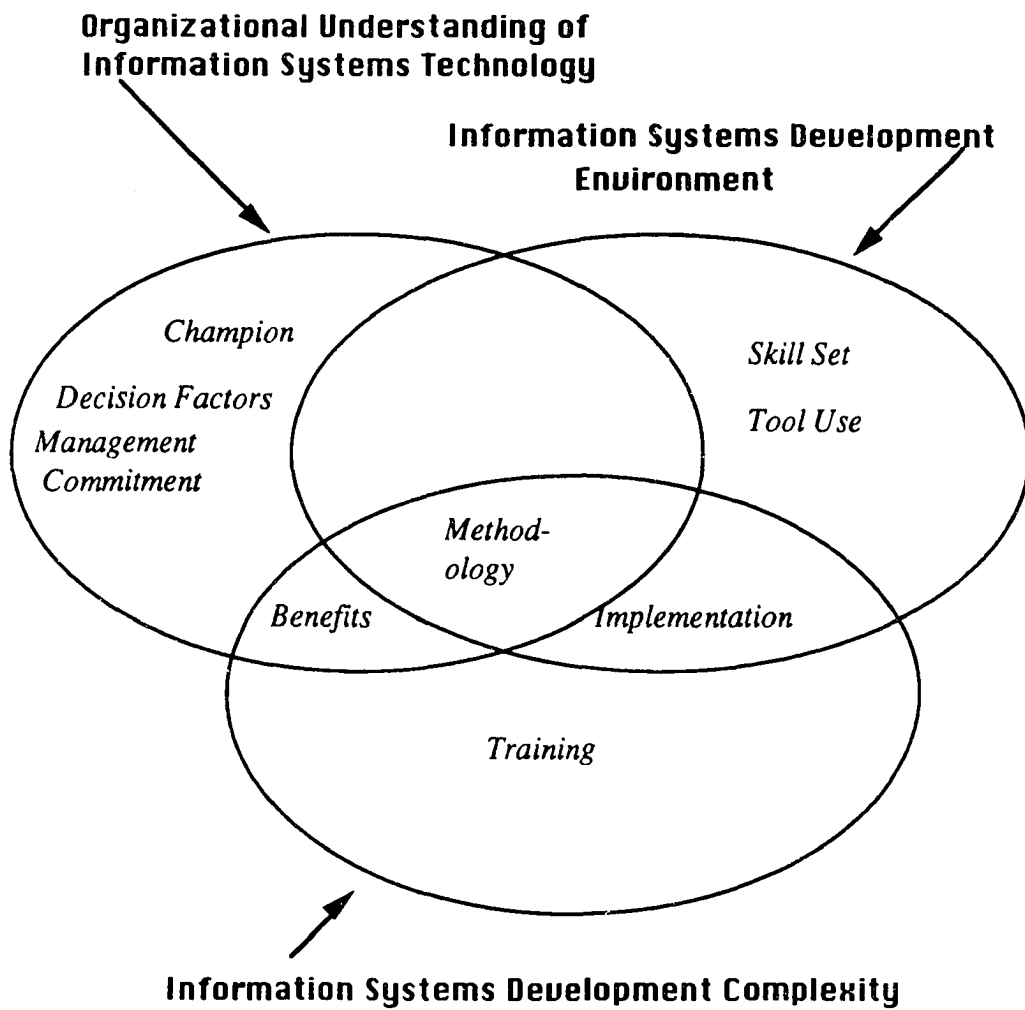


Figure 2

Core Themes and the Categories Related to Them



## Chapter Five: Conclusions and Suggestions for Future Research

### Conclusions

This study identified organizational and technological factors that contribute to successful implementation of CASE technology. The relationship among and between the factors was examined. The factors fall into three core categories: organizational understanding of information systems technology, information systems development environment, and information systems development complexity. This study found that successful CASE technology implementation involves a complex interaction between factors in each of these categories.

Organizational understanding refers to how well management understands the impact of CASE technology on the organization's information systems development activities. The role of the champion must be considered. A champion needs to have the authority to ensure that CASE technology will be successful. The benefits the organization expects to receive need to be balanced against the financial commitment the organization is willing to make. This category is linked to the category of information systems development complexity through the factor "benefits." The reasons the organization decided to adopt the technology must be clearly stated. These decision factors must be communicated to those involved in implementing the technology. The role of a methodology in the organization must be considered as a measure of organizational understanding of information technology. An organization that commits itself to a systems development methodology at all levels will better understand the role of information technology. The role of a methodology links all three categories.

Information systems development environment describes the context in which information systems are developed. This category includes the skill set of the information systems professionals in the organization, the way the tool is used, the implementation strategy followed, and the role of an information systems development methodology. Implementation strategy links this category with information systems development complexity.

Information systems development complexity is the degree of complexity of the systems that the organization develops using CASE technology. The factors in this category are benefits, implementation, methodology and training. The benefits category is also linked to organizational understanding; the implementation category is also linked to information systems development environment. If complex systems are being developed, training is of critical importance. Organizations that look at training as a dispensable item will not be successful.

#### Implications for IS Practitioners

This research indicates that, at a minimum, practitioners need to pay careful attention to: 1) how an information systems development methodology is used in their organizations, 2) how the tool itself will be used, and 3) how training will be conducted, including who will be trained. Each organization should also assess its level of understanding of information systems technology.

According to this research, the most important factor for CASE implementation success is the role of an information systems development methodology in an organization. Organizations developing information systems according to an information systems development methodology are more successful in their use of CASE technology. Therefore, practitioners

must be very disciplined in information systems development activities. An information systems development methodology should be followed, and the CASE tool used should support the methodology.

The use of a methodology spans all three categories of factors. In a structured information systems functional area developing applications using large systems, development teams must adhere to some type of approach. For smaller information systems functional areas, developing applications in one or two person teams, the presence or absence of a methodology is less critical. However, every organization's chances of CASE implementation success are greater with a methodology than without one.

This research also indicates that an organization planning to use CASE technology must determine how the tool will be used. The types of projects which will be conducted using the tool and the skill level of the information systems staff involved in these projects must be considered. An organization expecting the CASE tool to enforce an information systems development methodology will be disappointed. The organization must be willing to change both the way projects are managed and the composition of the project team. The focus must be on the quality of the deliverables. If reusability is an organizational objective, then projects must be conducted with reusability in mind. The project team must consider how models they are building can be used by other members of the organization.

An organization that chooses a tool that can be used across all phases of the systems development life cycle and uses the tool in that way must be prepared to invest heavily, at least at first, in training for all members of the organization who will be affected. This includes those who will be

directly affected because they will be using the tool, and those who will be affected indirectly because they will be using the outputs of the tool. Each organization needs to assess if theirs is the kind of organization that can be subjected to the rigid enforcement of a systems development life cycle methodology which such use would require.

Finally, each organization needs to determine its level of understanding of information systems technology. Effective use of CASE requires a different approach to information systems development. To make the best use of CASE technology, an organization must organize differently. The focus of the project team must be both short term and long term if the benefits of tool use are to be realized. Rewards need to be based both on the current project and the reusability of parts of the project for future projects. Technical support and enforcement of information systems development standards require that a support staff be put into place.

The designation of a champion of the new technology is less important. Organizations can succeed with or without a champion, but if one is used he or she must be chosen carefully and given the proper resources to have a positive effect on CASE technology implementation.

A finding of this research is that the use of a CASE tool for program source code generation is controversial. An organization planning to use ICASE for the entire life cycle should carefully assess how generated source code will fit into the organization.

#### Implications for IS Researchers

This research indicates that information systems researchers should continue to examine how information systems applications are being developed, especially as the technology changes and desktop computing

enables end users to develop more pieces of an information system. Much of the current research assumes that practitioners are developing information systems according to an information systems development methodology. Often this assumption does not apply at a particular site or for a particular project. Information systems professionals, facing increasing pressure to complete projects, often develop systems without adhering to generally accepted principles of information systems development. This disregard of sound practice should be studied to determine what price, if any, organizations pay for their increased attention to meeting today's deadlines and how those deadlines are set.

#### Suggestions for Future Research

This exploratory study identifies three categories of organizational and technological factors that contribute to the successful implementation of CASE technology. The focus of the present study is on ICASE tools. Additional research could be done to determine if other kinds of CASE tools are affected by these categories of factors.

## Appendix A: The Principles of Grounded Theory

### Introduction

This appendix explains the general principles of grounded theory, as discussed in Glaser and Strauss (1967), Strauss (1987) and Strauss and Corbin (1990). In particular, this appendix focuses on the terms particular to the approach. The intent is to define grounded theory as used in this research. This discussion is therefore limited.

#### What is a Grounded Theory?

The best definition of a grounded theory is provided by Strauss and Corbin "A grounded theory is one that is inductively derived from the study of the phenomenon it represents. That is, it is discovered, developed, and provisionally verified through systematic data collection and analysis of data pertaining to that phenomenon. Therefore, data collection, analysis and theory stand in reciprocal relationship with each other. One does not begin with a theory, then prove it. Rather, one begins with an area of study and what is relevant to that area is allowed to emerge" (Strauss & Corbin, 1990, p. 23). The term "grounded theory" is used both to describe the theory that emerges from the study, and the process by which the theory was developed, or in the words of Glaser and Strauss, "discovered." This qualitative analytical technique is very useful for theoretical development in areas where theoretical work is sparse. The information systems field is such an area.

The approach depends on the systematic analysis and collection of data, and relies heavily on a number of coding techniques. The next three sections of this appendix describe the coding techniques used in grounded theory. The last section describes theoretical sampling. The final section discusses the use of the approach in this research.

## Open Coding

Open coding is the first step in analyzing the data. The objective is to compile a list of categories into which the data fall, and to learn something about the dimensions and attributes of the categories.

### Definition

Open coding is defined as "the process of breaking down, examining, comparing, conceptualizing, and categorizing data" (Strauss & Corbin, 1990, p. 61). This is done by making comparisons and asking questions. Each piece of data is given a name, or a label. It is analyzed to determine its meaning. It is compared to other data so that data that represent the same concept are given the same names, or labels. The seed categories culled from the literature can be used to aid in the process of categorizing data, but often data for categories that are unexpected will be discovered in the analysis, giving rise to new categories.

### Example

In this research open coding was conducted using interview transcripts, interview notes, and field notes. Each note was examined and categorized. Initially this categorization was completed by annotating directly on the notes. From these notes a set of categories was compiled. Samples of the data from each of the categories are included in Appendix D. The following is a paragraph from an interview that will be used to illustrate the process. The sentences have been numbered for illustrative purposes only.

"(1) I think that there is also a feeling that the amount of code that you have to develop is probably a little more than what you want to be doing now. (2) For example, new tools like Visual Basic have a lot more behind the scenes...(3) You also have to do a large amount of coding for

screen scrolling. (4) We do spend a lot more time now in analysis, 50-50 where before 20-80 or whatever. (5) It does allow us to analyze the business and make changes in the later stages far more quickly than a conventional tool."

Sentence (1) was coded as Coding, sentence (2) was discarded as not being relevant to the research, sentence (3) was coded as Coding, sentence (4) was coded as Tool Use, and sentence (5) was coded as Benefits. To facilitate the process, a "code note" was written for each code. A code note is a note written by the researcher in which the properties of the category, as indicated by the data, are recorded and analyzed. Some of the code notes will be discarded as the analysis progresses, while others will be included in the final analysis and theory development.

This coding approach was used on all the data that was collected. The resulting categories were listed, and characteristics of the categories were noted.

### Axial Coding

Axial coding is the second step. Here the objective is to integrate the categories and the subcategories. Here the focus is on the development of a category in terms of the conditions that allow it to develop, the context in which it develops, the strategies by which it is managed, and the consequences of the strategies that are used. These specifics are the "subcategories" of the category.

### Definition

Again, referring to Strauss and Corbin, axial coding is defined as "a set of procedures whereby data are put back together in new ways after open coding, by making connections between categories. This is done by utilizing a coding paradigm involving conditions, context,

action/interactional strategies and consequences" (Strauss & Corbin, 1990, p. 96).

### Example

Once the categories were identified, code notes for each category were compiled and studied. The original list was pared down to nine categories, from 18. Dimensions and properties of each category were catalogued. The category "tool use" will be used to illustrate this process. All the code notes pertaining to tool use were compiled. An analysis of these notes revealed that tool use depends on the organizational context in the following ways. In organizations where an underlying methodology is enforced, the tool is used throughout the information systems development life cycle. In organizations where a methodology is not enforced, tool use is left up to the individual developer and varies depending on the individual's level of knowledge about the tool. The consequences of this mixed use are that tool use is greater in organizations where the methodology is enforced. Those organizations are more highly structured in the manner in which they develop information systems. The result is that the tool is more widely used, and used throughout more phases of the information systems development life cycle, in highly structured, or disciplined information systems development environments.

An analysis of the set of code notes for each of the remaining eight categories was conducted in a similar manner. For each category, a theoretical note was outlined.

### Selective Coding

The final step is to integrate the categories around a set of core categories. This is done through selective coding. The result of this

process is the development of the "story line" that is the descriptive narrative of the phenomenon under study, or the theory.

### Definition

Selective coding is "the process of selecting the core category, systematically relating it to other categories, validating those relationships, and filling in categories that need further refinement and development" (Strauss & Corbin, 1990, p. 116). In this research three core categories emerged from the data. These categories represent a higher level of abstraction of the original list of categories.

### Example

The list of nine categories was divided into groups of three in the following way. The first three categories were examined to determine which two were most alike and why. The first category was dropped, the fourth category was added and the process was repeated. Through this procedure of "triadic sorting" the similarities and differences among and between the categories were delineated. This revealed the underlying grouping and gave rise to the three core categories of information systems development environment, information systems development complexity, and organizational understanding of information technology. The categories that are contained in each core category describe that category, and the overlapping categories define the linkages between and among the categories.

### Theoretical Sampling

The data collection process is carried on in conjunction with the analysis when a researcher conducts a grounded theory study. Data collection is focused on the development and understanding of concepts.

Data is therefore collected to explicate the concepts that have been discovered while collecting data.

### Definition

Theoretical sampling is sampling on the basis of concepts. The concepts are the ones that have proven theoretical relevance to the emerging theory. Strauss and Corbin (1990) state that "proven theoretical relevance indicates that certain concepts are deemed significant because (1) they are repeatedly present or notably absent when comparing incident after incident, and (2) through the coding procedures they earn the status of categories" (Strauss & Corbin, 1990, p. 177).

### Example

The data collection began with a set of initial interviews and visits to ICASE sites. In these early interviews, many informants discussed the issue of source code generation. This topic was mentioned a number of times, and subsequently emerged as a category. Sites were then sought where source code generation was routinely conducted using the ICASE tool and from sites where source code generation was not performed using an ICASE tool. Data was then collected from these sites. This data collection continued until no new information regarding source code generation was being collected. The category was then deemed "theoretically saturated."

### Conclusions

An extensive discussion of the principles of grounded theory is beyond the scope of this work. This appendix illustrates the principles of open coding, axial coding, selective coding and theoretical sampling. Examples were chosen from the data collected for this research to illustrate how the techniques were applied to develop a theory of how organizational

and technological factors contribute to CASE implementation success. The reader is referred to Glaser and Strauss (1967), Strauss & Corbin (1990) and Strauss (1987) for further discussion.

## Appendix B

### The Questionnaire

Please describe your organization's experience with ICASE technology. For the purposes of this study, ICASE refers to any CASE tool which can be used in all phases of a systems development life cycle.

1. In general, how satisfied would you say your organization is with its implementation of ICASE technology?

Very satisfied   
  Somewhat satisfied   
  Neither satisfied nor unsatisfied   
  Somewhat unsatisfied   
  Very unsatisfied

2. Before the implementation of ICASE technology, systems were developed following a systems development methodology:

Always   
  Often   
  Usually   
  Never   
  Not sure

3. After the implementation of ICASE technology, systems were developed following a systems development methodology:

Always   
  Often   
  Usually   
  Never   
  Not sure

4. Before the implementation of ICASE technology, systems were developed adhering to documentation standards:

Always   
  Often   
  Usually   
  Never   
  Not sure

5. After the implementation of ICASE technology, systems were developed adhering to documentation standards:

Always   
  Often   
  Usually   
  Never   
  Not sure

6. After the implementation of ICASE technology, systems were developed using a code generator for database creation:

Always   
  Often   
  Usually   
  Never   
  Not sure

7. After the implementation of ICASE technology, systems were developed using a code generator for source code:

Always   
  Often   
  Usually   
  Never   
  Not sure

8. Was there an individual(s) who was designated to serve as a strong advocate or "sponsor" of the ICASE tool and thereby played a key role in its acquisition or use? (Check one)

Yes                       No                       Not sure

9. What was the implementation strategy used?

Pilot project                       Mandated use for all new projects                       Decision by project team describe \_\_\_\_\_                       Other, sure                       Not sure

10. Which of the following statements describes how training was handled in your organization? Please check **all** that apply.

Vendor hired for training                       Consultant hired for training                       "Trained the trainers"                       In-house training staff describe                       Other, describe

11. The following statements describe some reasons why organizations choose to implement CASE technology. Please indicate your impression of how important each objective is to your organization by circling the appropriate number.

Intended Objective	very important	somewhat important	not at all important		
To develop a better quality system.	1	2	3	4	5
To develop systems more quickly.	1	2	3	4	5
To improve documentation of developed systems.	1	2	3	4	5
To develop more maintainable systems	1	2	3	4	5
To improve productivity of systems personnel.	1	2	3	4	5
To enforce a systems development methodology	1	2	3	4	5
To enforce standards in systems development.	1	2	3	4	5
To generate source code automatically	1	2	3	4	5
Other, please describe.	1	2	3	4	5

12. Please indicate your impression of the degree to which each objective was realized by your organization by circling the appropriate number.

Realized Objective	fully realized	somewhat realized	not at all realized		
To develop a better quality system.	1	2	3	4	5
To develop systems more quickly.	1	2	3	4	5
To improve documentation of developed systems.	1	2	3	4	5
To develop more maintainable systems	1	2	3	4	5
To improve productivity of systems personnel.	1	2	3	4	5
To enforce a systems development methodology	1	2	3	4	5
To enforce standards in systems development.	1	2	3	4	5
To generate source code automatically	1	2	3	4	5
Other, please describe.	1	2	3	4	5

13. For how many years has your organization been using ICASE technology? \_\_\_\_\_ years

14. Our information technology organization is

centralized     decentralized     a combination of both

15. The ICASE tool the organization **initially** used was:

- Texas Instrument's IEF
- KnowledgeWare's IEW/ADW
- CGI System's PacBASE
- Arthur Anderson's Foundation
- Other-please name \_\_\_\_\_

16. The ICASE tool the organization is **currently** using (please check all that apply):

- Texas Instrument's IEF
- KnowledgeWare's IEW/ADW
- CGI System's PacBASE
- Arthur Anderson's Foundation
- Other-please name \_\_\_\_\_
- We are not currently using a CASE tool

17. In this organization, ICASE technology is being used to (please check only one):

- Develop new systems  Re-engineer existing systems
- Both to develop new systems and re-engineer existing systems

**Thank you for participating in this research!**

## Appendix C

### Analysis of Quantitative Data

#### Response Rate

Questionnaires were sent to members of the KnowledgeWare Users Group and the Metropolitan New York CASE Users Group (MNYCUG, subsequently disbanded). A request for participants was posted to a CASE users list on the Internet, and volunteers were mailed a questionnaire. Questionnaires were mailed or faxed to the researcher. Approximately 750 KnowledgeWare users and 330 members of the MNYCUG were mailed questionnaires. Ten questionnaires were sent to members of the CASE users list on the Internet. The total number of questionnaires distributed by all methods was 1090.

Table 4 summarizes response rate information. All responses received were usable responses. Only two questionnaires were returned missing responses to specific questions, but they were still usable questionnaires. Twenty-six questionnaires were returned, for a response rate of 2.4%. The Internet provided the highest response rates. The only people who were sent a questionnaire over the Internet were those who requested one. Apparently, once requesting a questionnaire respondents felt obligated to return it.

The mailing to members of the recently dissolved MNYCUG yielded the greatest number of responses, representing 50% of all responses received, even though this mailing was less than half the size of the KnowledgeWare User Group mailing. The MNYCUG mailing included a stamped addressed envelope while the KnowledgeWare User Group mailing didn't. This might account for the higher response rate among the first group. The president of the KnowledgeWare User Group concurred

with the expectation that most people would fax a response if a fax number was provided. This proved not to be the case. More responses were received through regular mail than in any other way.

One possible reason for the low response rate might be that some of the people who are members of user groups are not CASE users. The cover letter asked that the questionnaire be passed on to an appropriate respondent, but perhaps some people who received the questionnaire and were not users did not do so. Another possible reason for the low response rate is that the reasons for success are more complex than a questionnaire could allow for, although the form provided room for free-form responses. Finally, this population might be saturated with questionnaires and not willing to complete yet another.

While the response rate was low, the 26 responses do allow for some very preliminary analysis. The responses help to describe the organizations that are using ICASE. Limited inferences can be drawn from such a small sample, however.

### Analysis

The questionnaire data were analyzed in two ways. First, a correlation analysis was performed to determine the relationships between success and management commitment, technology infrastructure, technical and end user involvement, and implementation strategy. Then a correlation matrix was calculated to determine the relationship between the set of intended objectives for implementing CASE technology and the set of realized objectives for implementing CASE technology. The first part of this section discusses the response rate, followed by some discussion of descriptive statistics, including frequency distributions and means.

### Descriptive Statistics

The responses were divided into successful and unsuccessful groups based on respondents' answers to the question "In general, how satisfied would you say your organization is with its implementation of ICASE technology? (question 1)". Those who responded "very satisfied" or "somewhat satisfied" were classified as "successful," while those who responded "neither satisfied nor unsatisfied," "somewhat unsatisfied" or "very unsatisfied" were classified as "unsuccessful." Sixteen respondents, or 61.5%, were classified as "successful" and 10, or 38.5%, were classified as "unsuccessful."

The respondents were asked if ICASE was being used to develop new systems, re-engineer existing systems or to both develop new systems and re-engineer existing systems (question 17). Nearly 46% of the respondents are using ICASE to develop new systems. The same percentage is using ICASE to both develop new systems and re-engineer existing systems. Only 8% are using ICASE exclusively to re-engineer existing systems.

Respondents were asked about the use of a systems development methodology both prior to the implementation of ICASE technology (question 2) and after the implementation of ICASE technology (question 3). Prior to implementing ICASE technology, 19% "often" used a systems development methodology, 46% "usually" did, 31% "never" did, and 4% were unsure. After the implementation of ICASE technology, 31% "always" use a systems development methodology, 19% "often" use one, 42% "usually" use one, and only 8% "never" follow a methodology. These responses indicate that organizations that implement ICASE technology tend to also adopt an information systems development methodology, regardless of their prior experience with a methodology.

Respondents were also asked if a champion was designated for ICASE (question 8). Seventy-three percent of respondents indicated that a champion was designated, with the remaining 27% answering "no." The percentage of respondents who designated a champion is larger than the percentage of responses that were classified as "successful," indicating that some "unsuccessful" sites designated champions.

Thirty-eight percent of the respondents indicated that a pilot project alone was chosen as the implementation strategy (question 9), while 11% used a pilot project along with mandated use. Fifteen percent mandated use solely, 19% let the project team decide, and 15% used other methods (including single user implementation, implementation for only parts of a project, like data modeling, or by type of project).

The number of years that an organization has been using ICASE technology varied between 2 and 15, with nearly 31% using the technology for two years, another 31% using the technology for three or four years, 19% using the technology for five years, and the remaining 19% using the technology for more than five years.

#### Relationship Between Intended and Realized Objectives

Respondents were given a list of intended objectives and asked to rate their impression of the importance of the objective to their organization (question 11), from very important (1) to not at all important (5). They were also asked their impression of the degree to which each objective was realized by their organization (question 12), from fully realized (1) to not at all realized (5). The answers were then correlated to see how closely the intended objective matched the realized objective.

The highest correlation was between the intended and realized objectives of generating source code automatically. This was rated between

2 and 5 both for importance and degree realized. The correlation coefficient between the two was .76747. This may indicate that those organizations that intend to generate source code have a very clear focus about why they are implementing CASE technology. This focus enables them to realize their objectives. Nearly 58% of the respondents said that they "always," "often," or "usually" developed systems using a code generator for source code. About 35% said that they "never" used the source code generator.

The intended objective of developing a better quality system was rated between 1 and 3, while the realized objective of developing a better quality system was rated between 1 and 5. The correlation coefficient was .57521, indicating that these outcomes were somewhat correlated. The only other pair that correlated as highly was the intended objective of enforcing a systems methodology and the realized objective of enforcing a systems development methodology, both of which were rated between 1 and 5. The correlation coefficient for this pair was .56262. This is consistent with the increase in frequency of an organization following an information systems development methodology after ICASE technology was implemented, as discussed earlier.

The remaining pairs were not highly correlated at all. Developing systems more quickly rated between 1 and 4 as an intended objective, and received ratings of 1 through 5 as a realized objective. The correlation coefficient was only .11227. The intended objective of improving documentation and the realized objective of improving documentation were rated in the same way, and the correlation coefficient was only .31640.

The intended objective of developing more maintainable systems was rated between 1 and 3, indicating that it was very important to respondents,

yet the realized objective of maintainable systems was rated from 1 to 4. These two objectives had a very low correlation coefficient, of .26522. CASE technology alone will not assist organizations in developing maintainable systems. Changes in organizational practice regarding reuse, deadlines, and other aspects of systems development project management have a much greater impact on maintainability.

The intended objective of improving productivity of systems development personnel was rated very important (from 1 to 3) but the realized objective was rated 1 through 5. The correlation coefficient here was only .35779. Finally, the intended objective of enforcing standards in systems development was rated from 1 through 4, while the realized objective of enforcing standards was rated 1 through 5. The correlation coefficient was only .22469. Apparently following a systems development methodology doesn't include enforcing standards or improving documentation.

In summary, the generation of source code appears to be an area where both the intended objective and the realized objective are clearly defined. An organization can easily tell if it is generating source code or not, but an organization can not always tell if productivity has been improved. The only other pairs of objectives with correlation coefficients higher than .50 were those of enforcing a systems development methodology, and developing a better quality system.

Table 4  
Response Rates

Source	Number Sent	Number Returned	Response Rate
Internet	8	7	87.5%
KnowledgeWare	750	6	.08%
MNYCUG	350	13	3.7%

**Appendix D**  
**Interview Protocol**

**Interviewee:**

Name \_\_\_\_\_

Title \_\_\_\_\_

Organization \_\_\_\_\_

Address \_\_\_\_\_

Date \_\_\_\_\_

**Introduction**

The focus of this research is to determine some of the organizational and technical factors which contribute to successful computer-aided software engineering (CASE) technology implementation. This research will focus on upper CASE tools, also known as integrated CASE tools. Examples of commercial products which would fit this definition are KnowledgeWare's ADW, Texas Instruments' IEF and Arthur Anderson's Foundation.

**Success**

1. In general, how satisfied would you say your organization is with its implementation of CASE technology?
2. How long has your organization been using CASE?
3. Which tool did your organization originally purchase?
  - a. What factors influenced the decision to purchase this particular tool?
4. Which tool(s) is your organization currently using?
  - a. If the organization is currently using a different tool, or other tools in addition to the first tool purchased, what factors led to the purchase of another tool?
5. What were your organization's expectations with respect to CASE technology?
  - a. Budgetary expectations

- b. Time frame expectations
- c. Life cycle expectations
- 6. Have these expectations been met?
- 7. Why do you think they have been met (or have not been met)?
- 8. What do you think were the most important factors which contributed to your organization's success (or lack of success) in implementing CASE technology?
- 9. What types of projects are currently using CASE technology? (new projects, re-engineering projects, both)

#### Technology Infrastructure

- 10. Does the organization follow a formal systems development methodology?
  - a. If so, which one?

#### Technical Staff and End Users

- 11. Who did the training? (vendor, outside consultant, organization)
- 12. Approximately much was spent on training, per user of the tool?
- 13. How was the tool originally selected? (team, individual, who?)

#### Management Commitment and Support

- 14. Did you have a formal, written plan for the tool's implementation?
- 15. Was a champion designated?

#### Implementation Strategy

- 16. How was the tool implemented? (pilot project, other)
- 17. Is there anything else you think I should know about your organization and CASE technology?

If the interview has gone well:

- 18. Is there anyone else in your organization with whom you think I should speak?

19. Do you think your organization would be willing to participate in a field study of CASE technology implementation? If so, how could I arrange to get your organization's commitment?

## Appendix E

### Samples of Code Notes

#### Code Note: Benefits

Interview T: "I think that's about it. I can't overemphasize the practical nature. Balancing the long term benefits with the short term benefits. How do you justify such a huge investment, slowing down a project. That's why we're looking for more incremental approaches rather than all or nothing."

#### Code Note: Champion

Interview W: "The end user of this system wasn't someone who had the visibility to really champion it."

#### Code Note: Decision Factors

Interview T: "Bachman was chosen because it was stronger in the data modeling area. One component. We're not using one integrated tool because of strength and weakness in different areas."

#### Code Note: Implementation

Interview T: "We're trying to look at it as part of the overall deployment of the technology. We try the pilot, you learn what is going wrong or right, you add the training aspect to the whole process. And hopefully, that will get us through it."

#### Code Note: Management Commitment

Interview M: "There's gotta be a change in the way management incents [sic] employees."

"If they were to turn around and say, not only are you incented on making your deadline, you're going to be incented on how much reusability you've achieved from looking at the corporate models that are in existence and reusing the existing models, the existing routines out of

lib, whatever it is that took place. That's got to be a part of it. People have to be incented to do something or they're not going to do it. They're going to do what they're incented to do. Right now they're incented to make deadlines."

Code Notes: Methodology:

Interview Q: "I don't think it's the CASE tool. I believe it's discipline."

"And uh you know it's not so much that using the CASE tools but a lot of us are not disciplined in terms of structured approach, structured methodology."

Interview T: "We have some people using the traditional methodology but we've tended to implement one and then walk away. It doesn't fit here, doesn't fit there. We've gotten away from it. What we're trying to do-it's just as important when we look at this process leader team, in the work with the teams, the communication flow from the team thru the process leader that results in the modification of the process to add value there. Must recognize best practices and get it back into the methodology, our approach."

Code Note: Skill Set

Interview M: "I'll tell you another individual that I'm good friends with, actually runs his own consulting firm down in Atlanta. One of the lines that he says, "Great people can make poor tools work. Great tools can make poor people work."

Code Note: Tool Use

Interview W: "They also had this concept of just defining the business rules, press a button, and there was no coding."

Code Note: Training

Interview T: "Overall in general, nobody has enough money for training. Or time. We're not good at just-in-time training. Our process leader group is working with our training group to determine when is the right time for training."

Code Notes: Success

Interview Q: "It's like any other project we do. I truly believe it's like another project. Sometimes we're not successful and we pull the plug on projects and we sort of don't necessarily want to go back and consider what should it have really cost, this kind of thing. It didn't work and it's time to move on."

## Reference List

- Ahituv, N., Hadass, M., & Neumann, S. (1984). A flexible approach to information systems development. MIS Quarterly, 8(2), 69-78.
- Avison, D. E., & Fitzgerald, G. (1988). Information systems development: current themes and future directions. Information & Software Technology, 30(8), 458-466.
- Avison, D. E., & Wood-Harper, A. T. (1991). Information systems development research: An exploration of ideas in practice. The Computer Journal, 34(2), 98-112.
- Bachman, C. (1988, July 1). A CASE for reverse engineering. Datamation, pp. 49-50,54-56.
- Banker, R. D., & Kauffman, R. J. (1991). Reuse and productivity in integrated computer-aided software engineering: an empirical study. MIS Quarterly, 15(3), 375-401.
- Baroudi, J. J., & Orlikowski, W. J. (1989). The problem of statistical power in MIS research. MIS Quarterly, 13(1), 87-106.
- Baum, D. (1992, July 27). Can backlogs be beaten? Information Week, pp. 46,50.
- Bellin, D. (1991) Information commodities and the production process: how the introduction of CASE tools affects the SDLC. Unpublished doctoral dissertation, City University of New York.
- Benbasat, I. (1985). An analysis of research methodologies. In F. W. McFarlan (Eds.), The Information Systems Research Challenge (pp. 47-85). Boston: Harvard Business School Press.
- Benbasat, I., Goldstein, D. R., & Mead, M. (1987). The case research strategy in studies of information systems. MIS Quarterly, 11(3), 369-386.
- Bennett, K. H. (1991). Automated support of software maintenance. Information & Software Technology, 33(1), 74-85.
- Bessemer, J. (1991). CASE in the 90s-a look ahead (part II). CASE Trends, 3(8), 24-25, 30, 32, 34.

- Borland, R. J., & Hirschheim, R. A. (Ed.). (1987). Critical Issues in Information Systems Research. New York, N.Y.: John Wiley.
- Brancheau, J. C., & Wetherbe, J. C. (1987). Key issues in information systems management. MIS Quarterly, *11*(1), 23-36.
- Brancheau, J. C., & Wetherbe, J. C. (1990). The adoption of spreadsheet software: testing innovation diffusion theory in the context of end-user computing. Information Systems Research, *1*(2), 115-143.
- Brewer, J., & Hunter, A. (1989). Multimethod Research. London, England: Sage Publications.
- Brooks, F. P. (1987). No silver bullet: essence and accidents of software engineering. Computer, *20*(4), 10-19.
- Brown, M. (1991). Making CASE work-practical CASE adoption tips. CASE Trends, *3*(7), 24-27.
- Bruce, T. A. (1988). CASE brought down to earth. Database Programming and Design, *1*(10), 22-28.
- Burkhard, D. L. (1989). Implementing CASE tools. Journal of Systems Management, *40*(5), 20-25.
- Calloway, L. J. (1988) An approach for assessing tools for designing dialog structures: a study of the dialog charts. Doctoral dissertation, New York University.
- Calloway, L. J., & Arlav, G. (1990). Developing and using a qualitative methodology to study relationships among designers and tools. In H.-E. Nissen, H. K. Klein, & R. Hirschheim (Ed.), The Information Systems Research Arena of the 90's, I (pp. 415-433). Copenhagen, Denmark:
- Cameron, J. R., Campbell, A., & Ward, P. T. (1991). Comparing software development methods: Example. Information & Software Technology, *33*(6), 386-402.
- Campbell, D. T., & Stanley, J. C. (1963). Experimental and Quasi-Experimental Designs for Research. Boston, MA: Houghton-Mifflin.
- Case, A. F., Jr. (1986). Information Systems Development: Principles of Computer-aided Software Engineering. Englewood Cliffs, N.J.: Prentice-Hall.

- Chaney, C. (1992). Lincoln Benefit Life meets the market challenge with CASE. Journal of Systems Management, 43(1), 10-12,31.
- Chen, M., & Nunamaker, J. F., Jr. (1991). The architecture and design of a collaborative environment for systems definition. Data Base, 22(1-2), 22-29.
- Chen, M., Nunamaker, J. F., Jr. , & Weber, E. S. (1989). Computer-aided software engineering: present status and future directions. Data Base, 20(1), 7-13.
- Colter, M. A. (1984). A comparative examination of systems analysis techniques. MIS Quarterly, 8(1), 51-66.
- Cook, T. D., & Campbell, D. T. (1979). Quasi-Experimentation: Design and Analytical Issues for Field Settings. Chicago, IL: Rand-McNally.
- Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. Communications of the ACM, 31(11), 1268-1287.
- Danziger, M. R., & Haynes, P. S. (1989). Managing the CASE environment. Journal of Systems Management, 40(5), 29-37.
- Davis, G. B. (1988). Commentary: The hidden costs of end-user computing. Accounting Horizons, 2(4), 103-106.
- Davis, G. B., Lee, A. S., Nickles, K. R., Chatterjee, S., Hartung, R., & Wu, Y. (1992). Diagnosis of an information system failure: a framework and interpretive process. Information & Management, 23, 293-318.
- Davis, G. B., & Olson, M. H. (1985). Management Information Systems: Conceptual Foundations, Structure and Development. New York, New York: McGraw-Hill.
- DeMarco, T. (1978). Structured Analysis and Systems Specifications. New York, New York: Yourdon, Inc.
- Dennis, A. R., Burns, R. N., & Gallupe, R. B. (1987). Phased design: a mixed methodology for application systems development. Data Base, 18(4), 31-37.
- Dickson, G. W., Leitheiser, R. L., & Wetherbe, J. C. (1984). Key information systems issues for the 80's. MIS Quarterly, 8(3), 135-139.

- DiLorio, F. C. (1991). SAS Applications Programming: A Gentle Introduction. Boston, MA: PWS-Kent Publishing Company.
- Doke, E. R. (1990). An industry survey of emerging prototyping methods. Information & Management, 18(4), 169-176.
- Eliot, L. B. (1992). Critical success factors for CASE. CASE Trends, 3(9), 30,32.
- Eliot, L. B. (1992). Methodology and CASE: bridging the gap. CASE Trends, 4(1), 36-37.
- Eliot, L. B. (1992). Your measurement mileage may vary. CASE Trends, 4(4), 48-49.
- Feldman, J. (1990) Computer aided software engineering and the software development life cycle. Unpublished master's thesis, Baruch College, The City University of New York.
- Fenton, V. L. (1992) Helping novice users: a field study of cognitive style and student learning assisted by an unstructured full-text retrieval system. Unpublished doctoral dissertation, Stevens Institute of Technology.
- Fuggetta, A. (1993). A classification of CASE technology. Computer, 26(12), 25-38.
- Galliers, R. D., & Land, F. F. (1987). Choosing appropriate information research methodologies. Communications of the ACM, 30, 900-902.
- Gane, C. (1990). Computer-Aided Software Engineering. Englewood Cliffs, N.J.: Prentice-Hall.
- Gane, C., & Sarson, T. (1979). Structured Systems Analysis: Tools and Techniques. Englewood Cliffs, N.J.: Prentice Hall.
- Gavurin, S. L. (1991). Where does prototyping fit in IS development? Journal of Systems Management, 42(2), 13-17.
- Gibson, M. L. (1988, July 1). A guide to selecting CASE tools. Datamation, pp. 65-66.
- Gibson, M. L., Snyder, C. A., & Carr, H. H. (1990). Implementing a corporatewide information strategy through CASE. Journal of Information Systems Management, 7(3), 8-17.

Gibson, M. L., Snyder, C. A., & Rainer, R. K., Jr. (1989). CASE: Clarifying common misconceptions. Journal of Systems Management, 40(5), 12-19.

Glaser, B. G., & Strauss, A. L. (1967). The Discovery of Grounded Theory: Strategies for Qualitative Research. New York: Aldine Publishing Company.

Glasson, B. C. (1989). Model of system evolution. Information & Software Technology, 31(7), 351-356.

Gordon, C. L., Necco, C. R., & Tsai, N. W. (1987). Toward a standard systems development life cycle. Journal of Systems Management, 38(8), 24-27.

Gorry, G. A., & Scott Morton, M. S. (1971). A framework for management information systems. Sloan Management Review, 55-70.

Graham, D. R. (1989). Incremental development: review of nonmonolithic life-cycle development models. Information & Software Technology, 31(1), 7-20.

Guinan, P. J., Hopkins, N., & Coopriider, J. (1992). Automation of the applications development life cycle: measuring the value added. In Proceedings of the Thirteenth International Conference on Information Systems, Dallas, Texas.

Hackathorn, R. D., & Karimi, J. (1988). A framework for comparing information engineering methods. MIS Quarterly, 12(2), 203-220.

Hanna, M. A. (1990). Move is on to tie vision to information systems. Software Magazine, pp. 39-42,44-45.

Hawk, S. R., & Santos, B. L. D. (1991). Successful system development: The effect of situational factors on alternate user roles. IEEE Transactions on Engineering Management, 38(4), 316-327.

Hayley, K. J., & Lyman, H. T. (1990). The realities of CASE. Journal of Information Systems Management, 8(2), 18-23.

Henderson, J. C., & Coopriider, J. G. (1990). Dimensions of I/S planning and design aids: a functional model of CASE technology. Information Systems Research, 1(3), 227-254.

- Hildebrand, D. K., Laing, J. D., & Rosenthal, H. (1977). Analysis of Ordinal Data. Beverly Hills: Sage Publications.
- Hirschheim, R., & Klein, H. (1989). Four paradigms of information system development. Communications of the ACM, 32, 1199-1216.
- Huff, C. C. (1992). Elements of a realistic CASE tool adoption budget. Communications of the ACM, 35(4), 45-54.
- Humphrey, W. (1989). Managing the Software Process. Reading, PA: Addison-Wesley Publishers.
- Janson, M. A., & Smith, L. D. (1985). Prototyping for systems development: a critical appraisal. MIS Quarterly, 9(4), 305-316.
- Jenkins, A. M., Naumann, J. D., & Wetherbe, J. C. (1984). Empirical investigation of systems development practices and results. Information & Management, 7(2), 73-82.
- Jones, C. (1992). The business value of software-a new approach. CASE Trends, 4(7), 56-60, 70.
- Jones, R. (1986). Automated tools for the analyst. Data Processing, 28(5), 241-243.
- Kaplan, B., & Duchon, D. (1988). Combining qualitative and quantitative methods in information systems research: a case study. MIS Quarterly, 12, 571-588.
- Keppel, G. (1982). Design & Analysis: A Researcher's Handbook (2nd ed.). Englewood Cliffs, N.J.: Prentice-Hall International.
- Keppel, G., & Zedeck, S. (1989). Data Analysis for Research Designs. New York: W. H. Freeman and Company.
- Kervin, J. B. (1992). Methods for Business Research. New York: HarperCollins Publishers, Inc.
- Keyes, J. (1992). Team tools targeting development process. Software Magazine, 12(15), 45-56.
- Kievit, K.-A., & Martin, M. P. (1989). Systems analysis tools-who is using them? Journal of Systems Management, 40(7), 26-30.
- Kowk, L. K. L., & Arnett, K. P. (1993). Organizational impact of CASE technology. Journal of Systems Management, 44(3), 24-28.

- Kraushaar, J. M., & Shirland, L. E. (1985). A prototyping method for applications development by end users and information systems specialists. MIS Quarterly, 9(3), 189-198.
- Kydd, C. T. (1989). Understanding the information content in MIS management tools. MIS Quarterly, 13(3), 277-290.
- Lederer, A. L., & Nath, R. (1991). Managing organizational issues in information systems development. Journal of Systems Management, 42(11), 23-27,39.
- Lee, A. S. (1989). A scientific methodology for MIS case studies. MIS Quarterly, 13(1), 33-50.
- Lee, A. S. (1991). Integrating positivist and interpretive approaches to organizational research. Organization Science, 2(4), 342-365.
- Lee, J., & Kim, S.-h. (1992). The relationship between procedural formalization in MIS development and MIS success. Information & Management, 22(2), 89-11.
- Liebetrau, A. M. (1983). Measures of Association. Beverly Hills: Sage Publications.
- Lin, C., & Chung, C. (1991). End-user computing in a CASE environment. Journal of Information Systems Management, 8(2), 17-21.
- Lyytinen, K. (1989). New challenges of systems development: A vision of the 90's. Data Base, 20(3), 1-12.
- Mahmood, M. A. (1987). Systems development methods-a comparative investigation. MIS Quarterly, 11(3), 293-311.
- Marcus, M. L. (1983). Power, politics and MIS implementation. Communications of the ACM, 26(6), 430-444.
- Marcus, M. L., & Robey, D. (1988). Information technology and organizational change: causal structure. Management Science, 34(5), 583-598.
- Martin, J. (1990a). Information Engineering. Volume I. Englewood Cliffs, N.J.: Prentice Hall.

- Martin, J. (1990b). Information Engineering. Volume II. Englewood Cliffs, N.J.: Prentice Hall.
- Martin, J. (1990c). Information Engineering. Volume III. Englewood Cliffs, N.J.: Prentice Hall.
- Martin, J., & Finkelstein, C. (1981). Information Engineering. Savant.
- Martin, P. Y., & Turner, B. A. (1986). Grounded theory and organizational research. The Journal of Applied Behavioral Science, 22(2), 141-157.
- McClure, C. (1989). CASE is Software Automation. Englewood Cliffs, N.J.: Prentice-Hall International.
- McFarlan, F. W. (Ed.). (1984). The Information Systems Research Challenge. Boston, MA: Harvard Business School Press.
- McGaughey, R. E., & Gibson, M. (1993). The repository/encyclopedia: essential to information engineering and fully integrated CASE. Journal of Systems Management, 44(3), 8-11, 41-42.
- McGrath, F. (1993). Checklist for buyers of ICASE products. Computer, 26(11), 108-110.
- McKeen, J. D. (1983). Successful development strategies for business application systems. MIS Quarterly, 7(3), 47-65.
- Moore, G. C., & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation. Information Systems Research, 2(3), 192-222.
- Moran, R. (1992, February 17). The case against CASE. Information Week, pp. 29-30,32.
- Mumford, E., Hirshheim, R., Fitzgerald, G., & Wood-Harper, T. (Ed.). (1985). Research Methods in Information Systems. New York: North Holland.
- Naumann, J. D., & Jenkins, A. M. (1982). Prototyping: The new paradigm for systems development. MIS Quarterly, 6(2), 29-44.
- Necco, C. R., Gordon, C. L., & Tsai, N. W. (1987). Systems analysis and design:current practices. MIS Quarterly, 11(4), 461-478.

- Necco, C. R., Tsai, N. W., & Holgeson, K. W. (1989). Current usage of CASE software. Journal of Systems Management, 40(5), 6-11, 19.
- Niederman, F., Brancheau, J. C., & Wetherbe, J. C. (1991). Information systems management issues for the 1990s. MIS Quarterly, 15(4), 475-495.
- Nolan, R. L., & Wetherbe, J. C. (1980). Toward a comprehensive framework for MIS research. MIS Quarterly, 14(2), 1-13.
- Norman, R. J., Corbitt, G. F., Butler, M. C., & McElroy, D. D. (1989). CASE technology transfer: a case study of unsuccessful change. Journal of Systems Management, 40(5), 33-37.
- Norman, R. J., & Forte, G. (1992). Guest editor's introduction: automating the software development process: CASE in the '90s. Communications of the ACM, 35(4), 27.
- Norman, R. J., & Nunamaker, J. F., Jr. (1989). CASE productivity perceptions of software engineering professionals. Communications of the ACM, 32(9), 1102-1108.
- Nunamaker, J. F., Jr., Chen, M., & Purdin, T. D. M. (1991). Systems development in information systems research. Journal of Management Information Systems, 7(3), 89-106.
- Olle, T. W., Hagelstein, J., Macdonald, I. G., Rolland, C., Sol, H. G., Assche, F. J. M. V., & Verrijn-Stuart, A. A. (1988). Information Systems Methodologies: A Framework for Understanding. New York, New York: Addison-Wesley Publishing Company.
- Olle, T. W., Sol, H. G., & Verrijn-Stuart, A. A. (Ed.). (1986). Information Systems Design Methodologies: Improving the Practice. North Holland.
- Orlikowski, W. (1988) Information technology and post-industrial organizations: an exploration of the computer-mediation of production work. Doctoral thesis, New York University.
- Orlikowski, W. (1989). Division among the ranks: the social implications of CASE tools for systems developers. In Proceedings of the Tenth International Conference on Information Systems, (pp. 199-210). Boston, MA:

- Orlikowski, W. J. (1988). CASE tools and the IS workplace: findings from empirical research. In Proceedings of the 1988 ACM SIGCPR Conference, (pp. 88-97). College Park, Maryland: ACM Press.
- Orlikowski, W. J. (1993). CASE tools as organizational change: investigating incremental and radical changes in systems development. MIS Quarterly, 17(3), 309-340.
- Orlikowski, W. J., & Baroudi, J. J. (1991). Studying information technology in organizations: research approaches and assumptions. Information Systems Research, 2(1), 1-28.
- Rowe, J. M. (1993). Can enforced standardization affect CASE usage? Journal of Systems Management, 44(3), 29-33.
- Rubin, H. (1992). CASE success and software metrics: the "odd couple". CASE Trends, 4(2), 38,40,60.
- Ryan, T. F., & Bock, D. B. (1992). A socio-technical systems viewpoint to CASE tool adoption. Journal of Systems Management, 43(11), 25-29.
- Schatzman, L., & Strauss, A. L. (1973). Field Research: Strategies for a Natural Sociology. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.
- Schmitt, N. W., & Klimoski, R. J. (1991). Research Methods in Human Resources Management. Cincinnati: South-Western Publishing Company.
- Schneider, H.-J. (Ed.). (1979). Formal Models and Practical Tools for Information Systems Design. Amsterdam: North Holland.
- Scott Morton, M. (1985). The state of the art of research. In F. W. McFarlan (Eds.), The Information Systems Research Challenge (pp. 13-41). Boston: Harvard Business School Press.
- Seidman, I. E. (1991). Interviewing as Qualitative Research. New York: Teachers College, Columbia University.
- Slater, D. (1992). PacBase, IEF lead rising CASE satisfaction. Computerworld, 26(31), 81-82.
- Souza, E. (1990). Strategies for software engineering implementation. Journal of Information Systems Management, 7(3), 33-37.

Stone, J. A. (1993). Inside ADW and IEF: The Promise and Reality of CASE. New York, NY: McGraw-Hill, Inc.

Straub, D. W. (1989). Validating instruments in MIS research. MIS Quarterly, 13(2), 147-169.

Strauss, A., & Corbin, J. (1990). Basics of Qualitative Research: Grounded Theory, Procedures and Techniques. Newbury Park, Ca: Sage Publications.

Strauss, A. L. (1987). Qualitative Analysis for Social Scientists. New York, NY: Cambridge University Press.

Strauss, A. L. (1987). Qualitative Analysis for Social Scientists. New York: Cambridge University Press.

Swanson, E. B., & Beath, C. M. (1989). Reconstructing the systems development organization. MIS Quarterly, 13(3), 293-307.

Todd, M., Coleman, R. L., & Shimonek, J. M. (1991). A CASE perspective of systems development: Woodmen Accident and Centel. Journal of Systems Management, 42(9), 13-16,35.

Toraskar, K. (1990). How managerial users evaluation their decision-support?: A grounded theory approach. In H.-E. Nissen, H. K. Klein, & R. Hirschheim (Ed.), The Information Systems Research Arena of the 90's, II (pp. 435-463). Copenhagen, Denmark:

Turner, B. A. (1983). The use of grounded theory for the qualitative analysis of organizational behaviour. Journal of Management Studies, 20(3), 333-348.

Tyran, C. K., & George, J. F. (1993). The implementation of expert systems: a survey of successful implementation. Data Base, 24(1), 5-15.

Van De velde, R. (1992). CASE adoption at Imperial Oil. Journal of Systems Management, 43(9), 24-25, 28-29, 36,38.

Van Maanen, J., Dabbs, J. M., Jr., & Faulkner, R. R. (1982). Varieties of Qualitative Research. Beverly Hills, CA: Sage Publications.

Vessey, I., Jarvenpaa, S. L., & Tractinsky, N. (1992). Evaluation of vendor products: CASE tools as methodology companions. Communications of the ACM, 35(4), 90-105.

Weinberg, R. S. (1991). Prototyping and the systems development life cycle. Journal of Information Systems Management, 8(2), 47-53.

Weinman, E. D. (1992). Editorial: CASE & database markets in perspective. CASE Trends, 4(8), 6.

Wojtkowski, W., & Wojtkowski, W. G. (1990). A look at the status of CASE technology. In A. M. Jenkins, H. S. Siegle, W. Wojtkowski, & W. G. Wojtkowski (Eds.), Research Issues in IS: An Agenda for the 1990's (pp. 173-192). Dubuque, IA: William C. Brown.

Wynekoop, J. L., & Conger, S. A. (1990). A review of computer aided software engineering research methods. In H.-E. Nissen, H. K. Klein, & R. Hirschheim (Ed.), Information systems research: contemporary approaches and emergent traditions, (pp. 129-154). Copenhagen, Denmark: Elsevier Science Publishers, B.V.

Yadav, S. B., Bravoco, R. R., Chatfield, A. T., & Rajkumar, T. M. (1988). Comparison of analysis techniques for information requirement determination. Communications of the ACM, 31(9), 1090-1097.

Yellen, R. E. (1992). What do users really think about CASE? Journal of Systems Management, 43(2), 16,39.

Yin, R. (1984). Case Study Research: Design and Methods. Beverly Hills, CA: Sage Publications.

Yourdon, E. (1986). What ever happened to structured analysis? Datamation.

Yourdon, E. (1989). Modern Structured Analysis. Englewood Cliffs, New Jersey: Prentice-Hall.

Yourdon, E. (1990). Structured techniques in the 90s? Computer Language.

Yourdon, E. (1991). Modern Structured Analysis. Englewood Cliffs, N.J.: Prentice-Hall International.

Yourdon, E., & Constantine, L. (1979). Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design. Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Yuen, M. K., & Spurgeon, J. R. (1992). Why organizations aren't more successful with CASE. CASE Trends, 4(5), 24-26,31.