Georgia State University
Digital Archive @ GSU

Computer Information Systems Dissertations | Department of Computer Information Systems

1-11-2006

Upgrading Packaged Software: An Exploratory Study of Decisions, Impacts, and Coping Strategies from the Perspectives of Stakeholders

Huoy Min Khoo

Follow this and additional works at: http://digitalarchive.gsu.edu/cis_diss

Recommended Citation
Khoo, Huoy Min, "Upgrading Packaged Software: An Exploratory Study of Decisions, Impacts, and Coping Strategies from the Perspectives of Stakeholders" (2006). *Computer Information Systems Dissertations.* Paper 5.

This Dissertation is brought to you for free and open access by the Department of Computer Information Systems at Digital Archive @ GSU. It has been accepted for inclusion in Computer Information Systems Dissertations by an authorized administrator of Digital Archive @ GSU. For more information, please contact digitalarchive@gsu.edu.

PERMISSION TO BORROW

In presenting this dissertation as a partial fulfillment of the requirements for an advanced degree from Georgia State University, I agree that the Library of the University shall make it available for inspection and circulation in accordance with its regulations governing materials of this type. I agree that permission to quote from, to copy from, or publish this dissertation may be granted by the author or, in his/her absence, the professor under whose direction it was written or, in his absence, by the Dean of the Robinson College of Business. Such quoting, copying, or publishing must be solely for the scholarly purposes and does not involve potential financial gain. It is understood that any copying from or publication of this dissertation which involves potential gain will not be allowed without written permission of the author.

HUOY MIN KHOO

NOTICE TO BORROWERS

All dissertations deposited in the Georgia State University Library must be used only in accordance with the stipulations prescribed by the author in the preceding statement.

The author of this dissertation is:

Huoy Min Khoo
1944 Oak Glen
New Braunfels, TX 78132

The director of this dissertation is:

Dr. Daniel Robey, John B. Zellars Professor
Computer Information System Department
J. Mack Robinson College of Business
P.O. Box 4015
Atlanta, Georgia 30302-4015

UPGRADING PACKAGED SOFTWARE: AN EXPLORATORY STUDY OF DECISIONS, IMPACTS, AND COPING STRATEGIES FROM THE PERSPECTIVES OF STAKEHOLDERS

BY

HUOY MIN KHOO

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree

Of

Doctor of Philosophy

In the Robinson College of Business

Of

Georgia State University

GEORGIA STATE UNIVERSITY
ROBINSON COLLEGE OF BUSINESS
2005

Copyright by

Huoy Min Khoo

2005

ACCEPTANCE

This dissertation was prepared under the direction of Huoy Min Khoo's Dissertation Committee. It has been approved and accepted by all members of that committee, and it has been accepted in partial fulfillment of the requirements for the degree of Doctoral of Philosophy in Business Administration in the Robinson College of Business of Georgia State University.

Dean: Dr. H. Fenwick Huss

DISSERTATION COMMITTEE

Chair: Dr. Daniel Robey

Dr. Ephraim McLean

Dr. Mark Keil

Dr. Pamela Barr

TABLE OF CONTENTS

TABLE OF CONTENTS ..... **VI**

LIST OF FIGURES ..... **XII**

ABSTRACT ..... **XIII**

CHAPTER 1: INTRODUCTION **15**

1.1 Software Upgrade Problems 17

1.2 Importance of Software Upgrade Issues 19

CHAPTER 2: BACKGROUND LITERATURE REVIEW **22**

2.1 Information Systems Maintenance 24

2.2 Influences on Packaged Software Upgrade 25

2.2.1 Demographic Factors 26

2.2.2 Environmental Forces 27

2.2.2.1 Vendors 29

2.2.2.2 Partners 30

2.2.3 Internal Stakeholders 30

2.3 Coping Strategies for Packaged Software Upgrade 32

2.4 Impacts of Packaged Software Upgrade 35

2.4.1 High Level Tangible Measures 36

2.4.2 Single System Evaluation 37

2.4.3 Costs and Benefits Analysis in IT Implementation 38

CHAPTER 3: RESEARCH QUESTIONS **40**

CHAPTER 4: METHODOLOGY **43**

4.1 Qualitative Approach 43

4.2 Philosophical Foundation 44

4.3 Research Design 46

4.3.1 Conceptual Framework 47

4.3.1.1 Description of the Diagram 48

4.3.2 Case Study 49

4.3.3 Site Selection 51

4.4 Data Collection 53

4.4.1 Interview Protocol 55

4.5 Data Analysis 62

4.5.1 Coding 63

4.6 Interpretive Case Study Evaluation 66

CHAPTER 5: CASE NUMBER ONE: WINDOWS 2000 UPGRADE AT CP 68

5.1 Corporate Governance Process 71

5.2 General Policy on Managing Technology Changes 73

5.3 Influences on Operating Systems Upgrade 74

5.4 Influences on IT's Decision to Set Windows 2000 as a Corporate Standard 76

5.5 CP's Philosophy in Managing Technology Changes 78

5.6 CP's Decision to Upgrade to Windows 2000 79

5.7 Deployment 81

5.7.1 Windows 2000 Upgrade at Divisional Headquarters 82

5.7.1.1 Desk Side Support Personal Strategy 88

5.7.2 Deployment at the Manufacturing Facilities 93

5.8 Impacts 102

5.8.1 Impacts of Windows 2000 Upgrade on Users 102

5.8.1.1 Impacts from Upgrade Process 103

5.8.1.2 Impacts from Software Changes 104

5.8.1.3 Impacts from OS Upgrade 105

5.8.1.4 Impacts on Office Applications 107

5.8.1.5 Impacts on Switching of E-mail Application 109

5.8.1.6 Learning Curve 111

5.8.2 Impacts of Windows 2000 Upgrade on IT 114

5.8.2.1 Improved Operating Systems 115

5.8.2.2 Support Features – Net Meeting 116

5.8.2.3 Active Directory 117

5.8.2.4 Impacts on Peripherals 122

5.8.2.5 Standardization of OS 123

5.8.2.6 An Unexpected Impact - A Problem and a Blessing 125

CHAPTER 6: CASE NUMBER TWO: SAP UPGRADE AT CP 127

6.1 Background 127

6.1.1 Keeping up with SAP 128

6.2 Decision Making Process 129

6.2.1 General Policy on SAP Upgrade 130

6.2.1.1 No Beta Version 131

6.2.1.2 18 months Rule 131

6.2.1.3 Vendor Sunset Date 132

6.2.2 SAP 4.6C Upgrade 132

6.3 4.6C Deployment 138
6.3.1 IT Personnel Training 138
6.3.2 Issues of Upgrade 139
6.3.3 Upgrade Process 140
6.3.4 Upgrade Policy – No New Functionality 142
6.3.5 Testing 143
6.3.6 Issues Resolution 145
6.3.7 Training 146
6.3.8 Role of Communication 148
6.3.9 Internal Audit 149
6.3.10 Cutover 150
6.3.11 Post Implementation Support 151
6.3.12 Business Approach to Upgrade 152
6.4 Impacts of SAP Upgrade 155
6.4.1 Impacts of SAP Upgrade on Users 155
6.4.1.1 Learning Curve 157
6.4.2 Impacts of SAP Upgrade on IT 159
6.4.2.1 Training 159
6.4.2.2 Testing 160
6.4.2.3 Customization 161
6.4.3 General Advantages and Disadvantages of SAP Upgrade 162
6.4.3.1 The Effect of Software Changes 163
6.4.3.2 The Effect of Vendor's Delay in Providing Solutions 164

CHAPTER 7: DISCUSSION 166

7.1 Packaged Software Upgrade Decision 168

7.2 Motivating Forces 169

7.2.1 Internal Requirements 169

7.2.1.1 Manager's Philosophy 169

7.2.1.2 Company Policy 170

7.2.1.3 Business Needs 170

7.2.2 External Dependency on Software Vendor 171

7.2.2.1 Functionality of Software 172

7.3 Internal Resources 173

7.4 Interaction of Motivating Influences and Internal Resources 174

7.5 Coping Strategies 175

7.5.1 Planned Strategies 176

7.5.1.1 Coping Strategies for Issues in Packaged Software 177

7.5.1.2 Implementation 179

7.5.1.3 Learning Assistance 180

7.5.1.4 Users' Strategy 182

7.5.2 Coping Strategy for Unanticipated Events 183

7.6 Impacts 183

7.6.1 Impacts Originated from Implementation Process 184

7.6.2 Impacts of Product 185

7.6.3 Circumstantial Impacts 188

CHAPTER 8: CONCLUSIONS 189

8.1 Implications for Research 190

8.2 Limitations 192

8.3 Implications for Practice 194

8.4 Future Research Directions 195

REFERENCES 197

VITAE 208

LIST OF FIGURES

Figure 1 Conceptual Framework 48

Figure 2 Partial Table of Contents from the SAP Decision Document 65

Figure 3 Excerpts from Decision Document 66

Figure 4 Induced Model..... 166

ABSTRACT

UPGRADING PACKAGED SOFTWARE: AN EXPLORATORY STUDY OF DECISIONS, IMPACTS, AND COPING STRATEGIES FROM THE PERSPECTIVE OF STAKEHOLDERS

BY

HUOY MIN KHOO

AUGUST 18, 2005

Committee Chair: Dr. Daniel Robey

Major Academic Unit: Computer Information Systems

Packaged software is widely adopted and has become an integral part of most organizations' IT portfolios. Once packaged software is adopted, upgrades to subsequent versions appear to be inevitable. To date, research on packaged software upgrade has not received the attention that it warrants, as academic research continues to focus on initial technology adoption.

To explore this understudied yet important area, three research questions were proposed: (1) What influences the decision to upgrade packaged software? (2) How do stakeholders cope with software upgrade? (3) How does a packaged software upgrade affect stakeholders?

A qualitative research method was used to study the research questions. Two cases were conducted at a *Fortune* 500 company located in the Southeastern region of United States. The first case studied Windows 2000 upgrades and the second case studied SAP 4.6C upgrade. A theoretical model with six components was induced from the study; the components are decision, motivating forces, contingency forces, planned strategies, corrective actions, and impacts. Upgrade decisions are the outcome of interaction between motivating forces that can originate from internal and external environments, and contingency forces. A decision to upgrade will lead to both positive and negative impacts as experienced by users and IT groups. However, stakeholders' experiences differ according to the types of software and also their roles in the company. Two types of strategies were observed in the study: planned strategies and corrective actions. Planned strategies were used to tackle anticipated issues, and corrective actions were adopted to solve ad hoc problems when negative impacts arose. Both strategies can affect the final outcome of impacts. Finally, in the event a corrective action was used, there is a chance that it will become a permanent planned strategy.

CHAPTER 1

INTRODUCTION

New technologies are introduced to the market at a pace that makes it difficult for organizations to keep up. As AOL Time Warner Chairman Stephen M. Case put it, "I sometimes feel like I'm behind the wheel of a race car, one of the biggest challenges is there are no road signs to help navigate. And in fact ... no one has yet determined which side of the road we're supposed to be on." (Garten 2001)

With more and more businesses embracing computing technology in daily operations, packaged software has become an essential part of an organization's IT portfolio. According to a report by IDC, the packaged software market was already a 154 billion dollar business at the end of 1999 (IDC 2000). Packaged software is commercial software that can be bought off the shelf, like MS Windows operating systems, word processing packages and so on. There are many kinds of software packages, ranging from end user applications to database management systems to telecommunication protocols.

In an online survey conducted by IDC, software was divided into five categories: office and office extension tools; development tools; operating systems for clients; operating systems for servers; and ERP, supply chain management and sales support systems. One problem that the IDC classification presents is the occurrence of redundant entries in multiple categories. For example, MS Windows falls under both office extension and operating systems. Because of that, a simpler and more intuitive classification will be used to classify the software packages in this research. Packaged software will be divided into five general categories.

The first category is *end user application,* e.g., spreadsheet, word processing, and presentation software. These software packages are used mainly to process daily office tasks.

The second category is *development* software, e.g., Oracle database, Borland C++, and Delphi. These are tools that IS professionals use to develop applications. The third category is *operating system*, which serves as the "brain" of the computer. A few examples of different types of operating systems available are Linux, MS Windows, and MacOS. The fourth category is *telecommunication and networking software* that provides linkage to internal and external information systems, e.g., Novel Netware. Finally, the fifth category pertains to function-specific applications like ERP and Siebel Sales Systems.

As software companies respond to business needs (Paine 2000) and add new features to make the software perform better, new versions of software have been released to the market in frequent successions. In fact, Microsoft launched Windows XP in 2001. To the manager, it is a constant question of whether the current version of software is “obsolete," or outdated, and warrants an upgrade. The decision to upgrade sometimes is not in the hands of an organization. Unless they are licensed to have total code autonomy, an upgrade can be just a matter of timing (Paine 2000). The question then becomes: when should they upgrade?

How do managers decide if they should migrate to Windows XP? If managers felt that Windows 97 is still sufficient for their current needs, should they migrate to the latest version? What are the factors that will influence their decision? These are some of the questions that organizations ask every time a new version of a software package emerges from the market. Deciding whether and when to upgrade are not the only problems facing IS management. Upgrade usually carries with it unexpected consequences as will be explained more in the next section.

1.1 Software Upgrade Problems

Upgrade means "raise to a higher grade" - Oxford Dictionary. It is a word that people associate with having a better quality. Organizations usually upgrade to the latest version of software with the goal of achieving efficiency and improved functionality that the current software lacks (Paine 2000). There are implicit assumptions that the latest version of software has better features, offers more functions and will increase users' productivity. Ideally, a software upgrade will fix existing bugs and enhance software. Unfortunately, a software upgrade seldom works properly the first time it is executed.

Occasionally, a software upgrade can be associated with the term "chaos." For example, when a key piece of software was upgraded at NYSE, trading was halted because the subsystem that handled small trades was crippled by a glitch which subsequently halted all trading (Bray and Staff 2001). In another example, a system crash resulting from a routine upgrade of mainframe software at Automatic Data Processing Inc. led to at least five brokerages being unable to get real-time updates on the cost of online stock trades (Weiss 2000).

Even though upgrades do not cause catastrophes everyday, routine upgrades often cause unexpected problems. Many IT professionals can recount past experiences of struggling to get software up and running after a trivial upgrade. One IS professional recalled his past experience dealing with migration from NT4.0 to Windows 2000 Server: "I had no end of trouble with NT4.0, after my 3.51 servers had been working fine forever" (Talley and Mitchell 2000). While many think an upgrade is just a simple task of installing the next version of software, overlaying new code over old code (Paine 2000), upgrade to software can lead to many problems.

One of the problems is inherent in the software package itself. Most software packages contain known bugs that haven't been ironed out when they are released. Indeed, one out of

seven software development firms deliver code without prior testing (Minasi and Garde 1999). As the following example illustrates, upgrading buggy software can create great inconvenience especially when the vendor doesn't inform the client of the problem and doesn't have solutions to the problem (Minasi and Garde 1999; Manes 2001).

When a contributing editor of *PC World* tried to copy old mail files from Outlook Express 4.0 to version 5.5., a simple file migration turned out to be nightmare. After hours of struggle, he called up Microsoft's public relations department. After many more hours, a developer finally admitted that the import function in the new version "pointed to an area that doesn't necessarily exist." When asked for a remedy, the editor was told the service pack would be available "in the next couple of months" (Manes 2001).

Another cause for upgrade problems is vendors' modification of previous design logic, which renders the new version incompatible with other software currently installed. According to a report in *USA Today*, new Office XP will create incompatibility issues if installed on the Windows 95 operating system because the file format has been changed (Baig 2001). In addition, the new version will become incompatible with software on other peripheral components. For example, after a new operating system is installed on a personal computer, an upgraded version of the driver is needed for the current printer and other interconnected peripherals. Thus, upgrading a piece of software usually involves more than the targeted system.

Frequent upgrades can be frustrating to programmers and users especially when the upgrade is problematic. For IS professionals, many man-hours are spent on correcting the glitches that accompany the upgrade. The upgraded version that was supposed to bring performance improvements sometimes slows down users' productivity when the software does not work properly after the installation. Furthermore, the vendor sometimes makes significant

interface changes that can further complicate the problem. I recall a personal experience with a colleague, after installing the latest version of database modeling tools, my colleague and I endured significant productivity loss because the familiar interface has been greatly overhauled. While not much functionality improvement was felt on the new software, it was a struggle just to perform daily tasks. As my colleague put it, "I feel like I am taking one step forward and two steps backward." Although the above example provides past professional experience as evidence, it demonstrates a problem that is commonly encountered in software upgrade.

1.2 Importance of Software Upgrade Issues

The above section provides some anecdotal evidence on the problems and impacts caused by performing upgrades on packaged software. Now, I will attempt to show why it is a topic worthy of further study. First, it is unknown if organizations are paying attention to the software upgrade problem. Until the problem is salient, no positive action can be deployed to solve the issues. Second, software upgrade is a continuous problem. Once the initial version of software is installed, it is very likely that the software will be upgraded unless an organization decides to abandon it. Third, the actual impact of software upgrade is unknown. Even though anecdotal evidence has shown that the upgrade process is problematic, no formal investigation has been conducted to fully understand the extent of the problem. Fourth, it is unknown how organizations are coping with the problem of software upgrade. Fifth, if the phenomenon of software upgrade can be understood further, then a better strategy can be deployed to help in software upgrade decisions.

Overall, software upgrade is an issue that has largely escaped academic attention.

To date, most evidence of the problem and impact of software upgrade is gathered from trade journals. No empirical research has been found that investigates these issues. Traditionally, academic research has focused on studies related to information system implementation (Lucas, Walton et al. 1988; Alavi and Joachimsthaler 1992; Thong, Yap et al. 1996) and no studies were found to concentrate solely on package upgrade issues.

One of the reasons package upgrade has not captured academic attention could be that it has been overlooked as a trivial problem. In the case of package upgrade, the software to be upgraded already exists. Thus, problems like user acceptance and technology fit should no longer be issues in this case. Although installing a new version of software is similar to adopting a new technology, it will not incur as much business process change as adopting a brand new technology package because it is merely substituting the latest version of software for the previous version. However, as the problems presented in the above section illustrated, some unanticipated glitches do occur when new software is installed.

While software upgrade does not sound as interesting as implementation of a new system, it has important implications that warrant further research. The objective of this study is to investigate software package upgrade phenomena from three aspects: decisions to upgrade, strategies used to cope with upgrade, and impacts of upgrade. Three research questions are posed to investigate the phenomena.

RQ1: What influences the decision to upgrade packaged software?

RQ2: How do stakeholders cope with software upgrades?

RQ3: How does a packaged software upgrade affect stakeholders?

Each research question will be discussed in more depth later in the proposal. In the next section, literature background on fields related to the three research questions will be discussed. Then, the research questions will be discussed and a conceptual model will be used to demonstrate how the research questions fit into the overall process of software package upgrade. Following that, a research plan will be laid out to provide understanding of the philosophical assumptions and methodology chosen to achieve the goal of the proposed research.

CHAPTER 2

BACKGROUND LITERATURE REVIEW

As pointed out in the Introduction section, few studies have paid attention to packaged software upgrade issues. In this section, related literature in fields pertaining to three research questions will be discussed under three subsections: Influences on Packaged Software Upgrade (Section 2.2), Coping Strategies for Packaged Software Upgrade (Section 2.3), and Impacts of Packaged Software Upgrade (Section 2.4). Most of the literature background comes from the general IT implementation and IT investment fields because research to date has yet to focus solely on general package upgrade. Even though general package upgrade is positioned in a slightly different category, as explained in the following paragraphs, selective literature in IT investment and IT implementation nonetheless can provide some insights to the proposed study. Before moving to the background literature on each research question starting from Section 2.1, let's first take a look at the scope of this study.

Software upgrade occurs when a higher version of software is installed in place of the older version of software. Although one can consider moving from the previous version of Lotus 1-2-3 to the latest version of MS Excel as a type of upgrade, an example of upgrading similar function software from one vendor to a different vendor, for this study the upgrade will be limited to applications from same vendor.

In the past, the IS field has focused mainly on system implementation instead of packaged software. As Lucas pointed out, "A significant amount of the work to date has focused on the general problem of implementation for information systems; most of the systems in the literature were custom designed" (Lucas, Walton and Ginzberg 1988). Compared to system implementation, which usually involves business process change or major development effort,

general package upgrade does not impact business process as much as new technology implementation. One reason is that the software being upgraded is already in place. Even though vendors tend to add functionality and improve user interfaces with new packages, the essential functions of the latest version of software package do not change. Between 'same application, same vendor' upgrade and 'similar application, different vendor' upgrade, the first category of upgrade should incur even less disruption than the second category because the same application upgrade should preserve most of the familiar features of the previous version.

Lucas et al. (1988) classified packaged software into two categories: general purpose and dedicated. General purpose packaged software is a tool that IS professionals and users use to solve problems. Examples of general purpose packages are Microsoft Word, Excel, and PowerPoint. These are end user applications commonly found in Microsoft Office Suites. In addition, examples of general purpose packages for IS professionals are Java Café, Oracle database management system, and software that is used as a development tool in system implementation.

A dedicated package, by contrast, is one that provides a solution to a particular function like accounts receivable, order entry or production planning. In general, an organization that adopts dedicated software will have to modify its process or modify the software to fit the task on hand (Lucas et al. 1988). Examples of dedicated packages are ERP systems, human resource planning packages like PeopleSoft, and customer relationship management (CRM) software. Both dedicated and general purpose software packages are of interest to this study.

2.1 Information Systems Maintenance

A field related to packaged software upgrade is information systems maintenance. Swanson and Beath (1989) called IS maintenance "A subject apparently lacking in glamor." It is another field that hasn't gained much attention. "Historically, new system development has occupied the foreground of IS work and maintenance the background" (Swanson and Beath 1989). System maintenance has always been thought of as a “grunt” task that begins after a system is implemented. The problem of maintenance first came into public awareness in the 1970s when rapid growth of in-house custom-developed application systems during the 1960s and early 1970s continued to expand and need constant upkeeping (Swanson and Beath 1989). From a narrow view, system maintenance is defined as "keeping the system up and running" (Swanson and Beath 1989). In a broader sense, "maintenance incorporates all task components involved in sustaining operational information systems within organizations" (Swanson and Beath 1989).

Kim and Westin (1988) defined software maintenance as "The activities performed on software after the programs have been installed." At first glance, general package upgrade seems to be subsumed under IS maintenance. However, there are two major differences between general software upgrade and traditional IS maintenance. First, a general package is bought off-the-shelf whereas information systems referred in the maintenance literature are custom-designed. Second, a general package upgrade usually replaces the previous version of software with a new version, whereas system maintenance usually works on a targeted function of a system to remedy a performance problem. Basically, maintenance is about performing additional correction, modification, or improvement over an existing information system. Prior studies related to maintenance have only looked at custom-built information systems in which almost

any changes to the system require programming (Swanson and Beath 1989). In fact, programmers have been known to spend tremendous time trying to trace past written codes and program logic before they can even modify a single line of code (Littman, Pinto et al. 1987).

2.2 Influences on Packaged Software Upgrade

This section will provide background information pertaining to the first research question, which is what influences organizational decisions to upgrade. Traditionally, technology has been adopted for various reasons. One, it is adopted to gain competitive advantage (Harrison, Mykytyn et al. 1997). Watson et al. (1997) list using IS for competitive advantage as one of the international issues in IS management. Two, it is used to reduce cost. According to a survey conducted by Hinton and Kaye (1996), "the most popular reason for justifying IT investment focused on the issues of cost reduction." Three, it is invested to improve productivity. In one study, 50 executives were asked to state the types of social subsystem costs and benefits that are most and least likely used in making IT investment decision, and productivity improvement was the most cited social subsystem benefit (Ryan and Harrison 2000).

The examples above are all related to IT investment in general. In the case of packaged software upgrade, what is the key motivation for an organization to abandon the previous version and adopt the new version? Intuitively, packaged software will be upgraded when it no longer supports daily operations. Nevertheless, many organizations upgrade their software packages before those packages become totally obsolete. To date, it is unclear what motivates organizations to consider the option of upgrade. Unlike IT implementation, which deals mainly with new technology adoption, general package upgrade faces a problem that is slightly different. One reason is the software package has already been implemented and used in daily

operations. Many factors that are considered during new technology adoption, like appropriateness of technology and user acceptance, are less of a concern when making the decision to upgrade.

Nevertheless, no research has focused on when a package is considered obsolete and in need of an upgrade. In addition, many reasons can lie behind the decision to upgrade or not to upgrade a package. In the following subsection, three areas that can potentially exert influences on organizational decisions to upgrade packaged software are discussed. The first area is demographic factors, the second area is external environmental forces, and the third area is internal stakeholders.

2.2.1 Demographic Factors

Organizational IT decisions may depend on which industrial sector the organization belongs to. A survey of 50 CIMA members whose responsibility was to appraise IT investment, found that a majority of organizations, especially those in the financial and retail sectors, are willing to adjust their IT decisions in response to their competitors and trends in the industry. In addition, the IT adoption philosophy of a firm can determine if it is more likely to adopt cutting edge technology. Three types of IT adopters have been identified: exploiters or innovators, competitors or early adopters, and participant or effective/efficient followers (Maier, Rainer Jr. et al. 1997). On the one hand, for a firm that prides itself as an exploiter or innovator of technology, there is a bigger chance it will adopt the latest upgrade as soon as the new software is released in the market. On the other hand, a firm that considers itself as a follower of technology will probably wait to upgrade until the migration is absolutely necessary for daily operations. Thus, demographics of a firm could influence the decision and the timing to upgrade.

2.2.2 Environmental Forces

One common practice that organizations use to keep up with what is "out there" is environmental scanning (Maier, Rainer Jr. et al. 1997). It is carried out because management is increasingly recognizing the link between IT resources and business strategy (Choudhury and Sampler 1997; Maier, Rainer Jr. et al. 1997).

One means that is used widely to collect information on upcoming technology is by employees attending conferences and reading trade journals. Occasionally, organizations will rely on consultants to provide information (Lederer and Mendelow 1990; Maier, Rainer Jr. et al. 1997). Some organizations have dedicated groups of people whose entire job responsibility is to cope with new technologies. They identify relevant new technology, assess its usefulness, and make recommendations for its implementation (Lederer and Mendelow 1990; Maier, Rainer Jr. et al. 1997). However, it is uncertain whether an organization conducts environmental scanning to seek out new releases of software, or if the information comes from vendor-initiated promotion. Environmental scanning nevertheless shows how environmental forces can affect an organization's IT planning decision.

In one study, Lederer and Mendelow (1990) looked at how organizations are affected by environmental forces, especially changing technology, and identified five dimensions of environment: technology, government, competitors, customers, and users. They then studied how those dimensions created categories of problems for IS management and how IS management coped with those problems. To them, "the environment can be viewed either as a deterministic force to which organizations respond or as an interdependent entity which organizations try to modify" (p.206). Hence, two approaches that they used are trying to resolve the problems on their own and influencing the environment to prevent the problems (Lederer and Mendelow

1990). In the next paragraph, six types of problems created by five environmental forces will be presented, with each problem italicized.

According to Lederer and Mendelow (1990), changing technology causes uncertainties in new technology performance, hence, creating a *buy or wait* problem because management could not decide when to adopt. Also, new technology may have new specifications that could cause *incompatibility* with older systems. Finally, aggressive promotion of new technology by media could create a so-called *technology mania* problem that can blind IS professionals from seeking the right solution and instead opt for flashy new technology.

In addition, changes in government policy, pressures of competitors, and demands from customers and users can all cause *priority setting* problems for IS management. When a competitor suddenly decides to slash price on a key product, the organization that is forced to counter the promotion will have to modify programs to accommodate that. In addition, an organization in a highly regulated industry like banking has to follow the regulation set by government closely and change its IT policy accordingly.

As users become more sophisticated, they sometimes acquire hardware and software on their own, which could create *sloppy systems* that did not comply with IS department standards. To make things worse, some users have *unrealistic expectations* toward a system, which can cause disappointment and put extra demands on the IS department, especially when the system fails to deliver the promise.

Of the six types of problems created, *buy or wait* and *incompatibility* problems are most relevant to packaged software upgrades. When trying to determine whether to upgrade a package, an organization is usually faced with the decision to *buy* now or *wait* until later. In addition, the latest version of software can create an *incompatibility* problem with existing

systems. Another problem created by technology changes that can also influence the upgrade decision is *technology mania*. If an organization bought into the bells and whistles promoted by vendors without carefully examining the costs and benefits of entire packages, the organization could fall into the trap of *technology mania*. The same situation can apply to packaged upgrade if an organization simply moves on to the latest version of software the moment it is available without examining the necessity to upgrade.

2.2.2.1 Vendors

Although Lederer and Mendelow (1990) included vendor watching as one of the coping mechanisms to tackle the problem of technology change, they failed to consider the vendor as one of the dimensions that can induce change. As the following paragraphs explain, the vendor will try to influence the organizational decision to adopt new technology.

A marketing strategy that software vendors employ to influence the decision of customers is software pre-announcement, or "vaporware." The primary objective of vaporware is to inform the market on an upcoming software product. Other than customers, pre-announcement can affect vendors' industry partners, competitors, distributors, and sometimes vendors themselves (Hoxmeier 2000). For the vendors, pre-announcement has the benefits of positioning new products, forestalling customers from buying competitor's products, establishing intended price level, and signaling their commitment to keep up with new technology, to name a few. However, a vendors' reputation can be damaged if the promised product is not delivered and it can lead to potential litigation risks (Hoxmeier 2000).

When vendors communicate the intended product delivery date and anticipated features, it can affect customers' decision to upgrade and purchase. Some organizations will hold off

buying new software and wait for the upcoming version (Hoxmeier 2000). Unfortunately, if the vendors fail to deliver the products on the specified date, organizations will be left with no software and an unfulfilled plan.

Another reason an organization performs packaged software upgrade is because vendors cease support. When a piece of critical software is no longer supported by the vendor, an organization is forced to upgrade to the next version.

2.2.2.2 Partners

Besides vendors, business partners can influence management decisions. For example, EDI is a technology that when installed in one location, does not realize the full benefit. In order to extract significant payoff from the inter-organizational system, powerful organizations have been known to force partners to adopt EDI technology (Lee, Clark et al. 1999). In packaged software situations, a business partner could exert pressure and force an organization to upgrade its software to become compatible with the version that the partner is using.

2.2.3 Internal Stakeholders

Another influence that can affect upgrade decisions lies inside the organization. Even though Lederer and Mendelow (1990) included users as one of the five dimensions of environmental forces, they did not include other stakeholders. In this section, the influence of internal stakeholders will receive additional attention.

A stakeholder is defined as "any group or individual who can affect or is affected by the achievement of the organization's objectives" (Freeman 1984, p.46). Internal stakeholders are

important to this study because they are believed to have influence over organization's decision to upgrade. Rowley (1997) pointed out, "Those developing stakeholder theory have concentrated on classifying stakeholders into useful categories that provide an understanding of how individual stakeholders influence firms' operations." The current study will be investigated from a stakeholder perspective, as explained in greater detail in Chapter 4.

Following is an example of how characteristic of stakeholders can affect the decision to adopt new technology. In a study conducted by Thong (1999), CEOs who had more IT knowledge were more likely to invest in IT. Furthermore, the reason that an employee wants to upgrade packaged software could be a personal one. For example, he may be interested in learning the latest technology to enhance his skill set. According to Leana and Barry (2000), "Individuals seek stimulation and variety in their work in order to fulfill self-development needs and maintain interest in, and satisfaction with their jobs."

Mitchell, Agle, and Wood (1997) proposed three attributes to classify stakeholders and to look at "the degree to which managers give priority to competing stakeholder claims," which they termed stakeholder salience. The three attributes that stakeholders possess are power, legitimacy, and urgency. According to their proposition, managers will perceive a stakeholder to be more salient if he or she has all three attributes present. Even though most IS literature looks at individual stakeholders, Rowley (1997) believed that a firm responds to the interaction of influences from multiple stakeholders when making a decision.

Hence, internal stakeholders can exert influence over decisions to upgrade packaged software. One reason is that stakeholders have different backgrounds and possess different motivations, based on past working experiences, positions in the organization, and career goals. Furthermore, the motivation to upgrade depends on the role that the stakeholder plays in an

upgrade process, e.g., user, developer, system administrator, upper management. For example, a user may prefer to utilize the latest technology but the system administrator may be hesitant to execute an upgrade because of the amount of work that he has to perform. Thus, by understanding factors that influence each stakeholder decision, it will help to discover how those influences have on the final decision of upgrade or not upgrade.

2.3 Coping Strategies for Packaged Software Upgrade

Although technology changes are a problem that many organizations share, past research in IS has nonetheless focused on IT adoption, IT implementation, and IT use. Not many researchers have paid attention to the coping strategy that managers employ to counter the force of rapid technology changes. The second research question emphasizes the issue of coping strategies for package upgrade.

To date, only two articles were found dealing with coping mechanisms on information technology change. In the first study, Lederer and Mendelow (1990) interviewed 20 IS executives to understand how five dimensions of environmental forces created problems for IS management. (A brief overview of the dimensions and the problems is available in Section 2.1.2.) From the interviews, they discovered two categories of coping mechanisms that IS management used: mechanisms internal to the IS department and mechanisms external to the IS department. In the following paragraphs, each coping mechanism will be italicized.

Internal to IS departments, organizations employ formal *systems design* techniques like functional decomposition and data flow diagrams to facilitate constant modification needed during priority resetting. Also adopted within IS departments is having *formal administrative policies and processes* regarding new technology. For example, setting a policy to only approve

new technology adoption after others have used it guards against the technology mania problem. In addition, a formal acquisition process can cope with decisions to buy or wait. And, *vendor watching* is used to anticipate technology change by closely following vendor's activities.

External to the IS department, IS managers inform and educate internal users on various issues like communicating project accomplishments and failures and explaining system requirements through a *public relation*s strategy. Also deployed is *political action* to influence government and legislators in their policy making. Another method used to cope with the problem is to *transfer the problem* outside of the local IS department. This strategy is usually used to cope with the priority resetting problem. For example, IS management that adopts packaged software will wait for the vendor to solve some of their problems. Furthermore, the local IS department can wait for the corporate IS department to share information with them. Finally, one additional mechanism is "muddle through," which happens when management treats the changing environment as "a fact of life" and deals with it on an emergency basis.

In the second study, Benamati and Lederer (2001) created a 34-item survey that contained general coping mechanisms developed from interviews with 16 IT professionals. Two examples of survey items are *coordinate communication among multiple vendors* and *attend conferences to keep informed of available new IT*. The surveys were sent to 1,000 IT professionals inquiring about extent to which those mechanisms were used in coping with rapid IT changes and how successful each mechanism was in helping IS professionals cope with the problem of IT change. From the 246 returned surveys, they identified five categories of coping mechanisms used by organizations*: education and training, internal procedures, vendor support, consultant support,* and *endurance*. Of the five categories, *education and training* is used most extensively. Five categories of coping mechanisms ranked in the order of popularity are:

Education and training: Stay informed of new IT as it becomes available and instruct or provide guidance in the use of new IT.

Endurance: Ignore or work around problems, and learn new IT without formal education.

Internal procedures: Develop processes to aid in the evaluation, acquisition, and implementation of new IT.

Vendor support: Rely on IT suppliers for problem determination and resolution, customization to, interfaces with, and functional enhancement to new IT.

Consultant support: Engage external IS professionals to help plan for, implement, problem solve, or provide ongoing support for new IT (Benamati and Lederer 2001).

Even though both research studies proved to be valuable, they only surveyed the opinions of IS professionals. Furthermore, the coping strategies pertain to general technology changes because the studies conducted did not identify the type of technology being surveyed. Coping with general software upgrades is not entirely the same as coping with technology changes in general. One known approach used by system administrators to cope with constant "sub" version upgrades is to conduct a release control. And a group of IS professionals have petitioned vendors to lengthen the period between the releases of new versions of software (McLean, 2001).

Furthermore, system administrators who have to implement upgrades may experience different problems than users who have to use the upgraded software. For this study, coping

mechanisms will be probed from multiple stakeholders based on different types of software. This should allow me to better understand how each group copes with the technology changes.

2.4 Impacts of Packaged Software Upgrade

This section of the literature review pertains to the third research question, which deals with the impacts of upgrade. With frequent release of packaged software, it has become crucial that an organization understand the effects of package upgrade on stakeholders. Upgrade impacts have been difficult to determine because the installation of the latest version of software can incur both positive and negative impact on organizations. People have always assumed that a new version of software has better features and will increase their work performance. However, installing the latest version of packaged software may lead to other problems that could potentially affect stakeholders' productivity. By understanding how an upgrade impacts stakeholders, management can evaluate the pros and cons of the upgrade, and provide a better strategy to cope with upgrade issues. Unfortunately, the effects of package upgrade have not been investigated by the IS research community.

Prior studies in the IS field have looked at IT impacts from different aspects. Some examined IT evaluations as an investment (Weill 1992; Barua, Kriebel et al. 1995) and others focused on single system evaluation (Belcher and Watson 1993; Finlay and Mitchell 1994). Moreover, two common types of measures used are tangible and intangible. For the purpose of this study, selective examples from three categories of IT evaluation will be included in the following section to provide a glimpse of the measures used. For each category, relevancy of the measures with respect to impacts of upgrade will be discussed. The first type of study adopts an organizational view of performance evaluation using tangible measures like accounting figures.

The second type of study focuses on impacts created by a single system implementation. A third type of study looks at the need to balance costs and benefits when evaluating an IT project.

2.4.1 High Level Tangible Measures

Traditionally, IT evaluation has focused on benefits of new technology and relied upon tangible measures. For example, when investigating the relationship between IT investment and firm performance, Weill (1992) used sales growth, return on assets (ROI), and labor productivity to measure firm performance. In another example, Hitt and Brynjolfsson (1996) used productivity, business profitability, and consumer surplus as three different measures in an economic modeling approach. In the third example, Barua, Kriebel, and Mukhopadhyay (1995) used a two-stage model to measure economic impact of IT. Some of the performance measures that they used were capacity utilization, inventory turnover, market share, and return on assets.

Even though high level accounting data like ROI, sales growth, business profitability, and market share are commonly used to assess the overall impact of an organization's IT investment, those types of measures may not be suitable to evaluate the impact of a software package upgrade for several reasons. First, it would be difficult to quantify the percentage of sales growth that is attributed to moving from one version of packaged software to the latest version. Second, it is doubtful that organizations actually capture information to measure relative improvement of package upgrade. Until the effect of a specific package upgrade can be separated from overall IT investment, the economic approach is deemed undesirable for this study, which is not interested in the overall impact of IT on organizations. Instead, the research interest is focused on general package upgrade, which only takes up a part of the overall IT budgets.

2.4.2 Single System Evaluation

Another type of research that is popular in the IS field is to examine the effect of a newly introduced single information system to determine if it is beneficial to a specific organization. Belcher and Watson (1993) collected usage statistics and interviewed key users to examine the cost and benefit of an executive information system (EIS), and to determine whether the EIS was being used the way it should be. The EIS was found to improve productivity by reducing staff time needed to update routine data, reduce the cost of distributing hard copy documents, and save on paid services by allowing employees to directly query the database. However, claimed improved decision making was perceived skeptically by management because it could not be quantified easily. In terms of intangible benefits, the EIS was found to connect employees throughout the organization and make them more informed. The main costs incurred were expenses associated with maintaining EIS and indirect expenses for business unit personnel who performed EIS-related tasks (Belcher and Watson 1993).

This approach allows the researcher to examine the impact of a specific system as a result of technology adoption. The idea of single system evaluation is deemed more appropriate for investigating impacts of upgrade because different software packages have different functions that could affect the outcomes of an upgrade. The goal of the proposed research is to understand how each type of package upgrade affects the organization.

In the next example, a single system evaluation is examined from two stakeholder perspectives. Finlay and Mitchell (1994) investigated the perceptions of benefits from the introduction of computer-aided software engineering (CASE) tools into a large British manufacturing organization. They compared developers' and customers' perception with objective measures and looked at intangible outcomes. They discovered that CASE tools

increased both development productivity and systems quality. The objective measure of productivity was function points. The subjective measures were obtained by asking developers and customers to rate their perceptions of progress toward the targeted goal. Both developers and customers perceived improvement in productivity of IS development, quality of output, and delivery time. However, developers had perceptions of higher improvement than the customers because customers were less concerned with productivity gains than developers' ability to fulfill the contract.

As Finlay and Mitchell's (1994) study shows, different stakeholders can potentially have different perceptions of outcomes in an IT project. Hence, it is important that a study includes different perspectives. For the current study, impacts of upgrade were probed from multiple stakeholder perspectives to get different views of how package upgrade affects each party.

2.4.3 Costs and Benefits Analysis in IT Implementation

The implicit assumption of IT implementation is usually that undertaking a project will result in improvement in the organization. As the following example will show, improvement depends on the organizational context and situation. Goodhue, Wybo, and Kirsch (1992) found that, while the public has assumed that organization-wide data integration is usually beneficial, data integration efforts may not be desirable in every situation because they do not provide sufficient benefits to offset their costs in certain organizational contexts. Three main organizational factors that can influence the impact of data integration are: the interdependence of subunits, the need for locally unique or flexible action by subunits, and the difficulty of designing and implementing systems with integrated data.

Data integration can be very useful in sharing information between sub-units in the organization and helping decision making throughout the organization. However, the authors argue that having a common logical design may work against unique local needs. If the sub-units within the organization are diverse, standardization may incur compromise costs on them. In addition, there might be a bureaucratic delay in getting changes to data models approved by sub-units. Thus, Goodhue et al. (1992) suggested adopting a partial integration model instead of full integration.

As the above example illustrates, IT adoption does not necessarily bring positive impacts onto the organization. In the case of packaged software upgrade, it is crucial not to assume that upgrades always bring better functionality and performance. The problems caused by upgrades may outweigh the benefits in some situations. Thus, the impacts of upgrade need to be carefully examined before making an upgrade decision and also tracked after implementing an upgrade to fully comprehend the extent of the impacts.

CHAPTER 3

RESEARCH QUESTIONS

I am interested in understanding the phenomenon of package software upgrade, how the constant change of technology affects organizational stakeholders in terms of their decision to upgrade or not to upgrade a software package when a new version is introduced. How do those decisions impact different stakeholders? In addition, how do stakeholders cope with upgrades? An exploratory study was used to study the phenomena through three research questions.

RQ 1: What influences the decision to upgrade packaged software?

The objective of the first research question is to explore what factors influence stakeholders' decisions to go ahead with new software or stay behind. It is critical to understand this issue because there can be many reasons why an upgrade is pursued, and those decisions can have impacts on the outcome of an upgrade as well as how stakeholders cope with it.

RQ2: How do stakeholders cope with software upgrades?

The purpose of the second research question is to explore the different coping mechanisms used by different stakeholders to lessen the negative effects induced by package upgrades. It is unclear how stakeholders cope with constant upgrades. Thus, it is crucial to understand whether organizations employ policies to attenuate the effects that they experience. In the absence of policy, do stakeholders adopt any tactic to counter the problems of software upgrade?

RQ 3: How does a packaged software upgrade affect stakeholders?

The goal of the third research question is to investigate how an upgrade affects stakeholders. This question is important because an organization may not be aware of what individual stakeholders feel about an upgrade. An upgrade that is trivial may be causing frustration to some users but is unknown to top management. Because each stakeholder has a different motivation, each may perceive the impact differently. Thus, it is critical to assess the impact from multiple perspectives to get a holistic view of its impact. This question is also important because only by understanding the positive and negative impact, will management be able to understand the tradeoff between keeping the older version and upgrading to the latest version. In many cases, upgrades may not be an option, e.g., when a vendor decides not to support an older version. But in other cases, bringing to awareness the impact that an upgrade has on stakeholders and their work can inform a better strategy for upgrades where they are possible. For example, many have assumed that the time spent on struggling with new features is a necessary part of the learning process during transition period (Ryan and Harrison 2000).

However, such struggles may be minimized if software upgrades are planned with awareness of stakeholder perceptions.

Also, software upgrades usually promise improved functionality. However, are those new features truly necessary to the users? Do users truly experience them? Is the time spent on installation, bug correction, constant learning, and work delay worth the benefit of having those features? How important are those features to users? These are some of the questions that subsume under research question number three.

All of the research questions are probed from multiple stakeholders' perspectives. It is important to do so because many parties are affected throughout the process of software upgrade. Even though past research on IT has focused on top management (Benamati and Lederer 2001) and information systems professionals (Lederer and Mendelow 1990), it is important that the research investigates the phenomena from the perspective of multiple parties who are involved in the software upgrade process. Their perceptions are equally important to understand various issues of software upgrade. At a minimum, stakeholders include the manager who decides to upgrade, the IS professional who is responsible to execute the decision, and the user who has to use the software package. However, there can certainly be more than three mentioned stakeholders, depending on the organizational structure and type of upgrade projects being carried out. Thus, the number of stakeholder groups will evolve inductively as the field study progresses.

CHAPTER 4

METHODOLOGY

4.1 Qualitative Approach

To investigate the aforementioned research questions, a qualitative approach was deemed most appropriate for several reasons. First, the research phenomena could only be studied in organizations where software upgrades were conducted. Unlike some phenomena that can be simulated and studied in a laboratory environment, packaged software upgrade issues cannot. According to Denzin and Lincoln (1998),

> "Qualitative research is multimethod in focus, involving an interpretive, naturalistic approach to its subject matter. This means that qualitative researchers study things in their natural settings, attempting to make sense of, or interpret, phenomena in terms of the meanings people bring to them."

Furthermore, according to Lee, Mitchell et al. (1999), qualitative research has four defining characteristics. First, it occurs in natural setting. Second, it derives data from the perspective of participants. Third, qualitative design can be flexibly changed to accommodate the demands of research situation. Fourth, it does not have standard instrumentation, which makes common notions of control, reliability, and validity difficult to obtain. The goal of this research is to study software upgrades where they occur and to understand how organizations react to them.

Second, I was interested in exploring the research questions from the perspective of stakeholders. An organization is made up of multiple stakeholders, each having a different role in daily business operations. Depending on the roles that stakeholders play, they will have different

opinions on why upgrade is necessary, and will have different perceptions on how upgrades impact them. It is important to understand stakeholders' past experience in dealing with packaged software upgrades. Probing multiple stakeholders provides different viewpoints and leads to more complete findings on upgrade decisions, impacts, and coping strategies.

Consistent with this exploratory research objective, I used a flexible method that would allow me the capability to follow up on interesting phenomena. A flexible data collection method that utilizes open-ended and semi-structured interview questions was favored to investigate the issues. It was also helpful to observe body expressions during the interviews and probe the issues further when an interesting question comes up. Thus, I chose a flexible data generation method that was sensitive to the social context in which data was produced (Mason 1996).

4.2 Philosophical Foundation

All qualitative research is guided by a "basic set of beliefs that guides action" (Guba, 1990, p.17) quoted from (Denzin and Lincoln 1998). Those beliefs are basic philosophical assumptions regarding a researcher's ontological and epistemological assumptions. According to Burrell and Morgan (1979), ontological assumptions "concern the very essence of the phenomena under investigation" whether "*reality* is given *out there* in the world, or the product of one's mind." Epistemology deals with "how one might begin to understand the world and communicate this as knowledge to fellow human beings." It is about the "grounds of knowledge" (Burrell and Morgan 1979).

Orlikowski and Baroudi (1991) classified IS research into three categories according to epistemological assumption: positivist, interpretive, and critical studies. Historically, the IS field has closely followed positivism (Orlikowski and Baroudi 1991), but lately, there is sign that

interpretive research is gaining popularity (Lee 1991; Orlikowski and Baroudi 1991; Walsham 1995).

Studies that fall under the positivist category are those that manifest quantifiable measures of variables, formal propositions, hypotheses testing, and the ability to draw inferences from a sample to a population. These studies are usually used to test theory (Orlikowski and Baroudi 1991). Ontologically, positivist researchers believe that there is an "objective physical and social world that exists independent of humans, and whose nature can be relatively unproblematically apprehended, characterized, and measured" (Orlikowski and Baroudi 1991).

On the other hand, critical studies usually aim to “critically evaluate and transform the social reality under investigation” (Orlikowski and Baroudi 1991). A critical researcher believes that there is a “dialectical relationship between elements and the totality” and that events cannot be studied in isolation from the social conditions that surround them (Orlikowski and Baroudi 1991).

Unlike positivist and critical studies, interpretive research assumes that "people create and associate their own subjective and intersubjective meanings as they interact with the world around them" (Orlikowski and Baroudi 1991). According to Burrell and Morgan (1979), the interpretive paradigm attempts to understand the "fundamental nature of the social world at the level of subjective experience." When adopting an interpretive paradigm, the researcher "seeks explanation within the realm of individual consciousness and subjectivity."

The study of software upgrades is new in the IS field, and not many established theories have been used to guide research. The objective of this research was to understand the phenomena in the natural setting from the perspective of those involved. It is believed that the phenomena do not exist "out there" and can only be understood by understanding the subjective

experiences of those affected by the decision. By acknowledging that reality is experienced differently by stakeholders, an interpretive paradigm expressly considers multiple subjective realities.

4.3 Research Design

"Research design is the logical sequence that connects the empirical data to a study's initial research questions and ultimately, to its conclusions" (Yin 1984). The first question that one needs to ask is how tight the preliminary research design should be, that is, how much planning should go into the preliminary design. For example, should a design contain a conceptual framework, or a set of research questions prior to entering the field? A research design that is tight in structure is one that has many pre-determined guidelines.

Several factors come to mind while trying to answer this question. First, it depends on the purpose of the research: whether the research is trying to explore an understudied phenomenon, to induce a social theory, to confirm well-defined constructs, or to test hypotheses. Traditionally, a loosely structured research design is associated with exploratory, inductive research whereas a tightly structured research design is linked to confirmatory, theory testing type of research. As Miles and Huberman (1994) pointed out:

> "Much qualitative research lies between these two extremes. Something is known conceptually about the phenomena, but not enough to house a theory. The researcher has an idea of the parts of the phenomenon that are not well understood and knows where to look for these things - in which settings, among which actors. And the researcher usually has some initial ideas about how to gather the information. At the outset, then, we usually

have at least a rudimentary conceptual framework, a set of general research questions, some notions about sampling, and some initial data-gathering devices." (p.17)

The next section shows the initial conceptual framework that guided the research design.

4.3.1 Conceptual Framework

"A conceptual framework explains, either graphically or in narrative form, the main things to be studied" (Miles and Huberman 1994). Those "main things" include key elements of research, and the presumed relationships among them. In this case, the conceptual framework does not represent a theoretical model that awaits confirmation. It is merely a map of those elements that were being explored in this study. The research results provided support for a different theoretical model, presented in Chapter 7. The preliminary conceptual framework is presented in Figure 1.

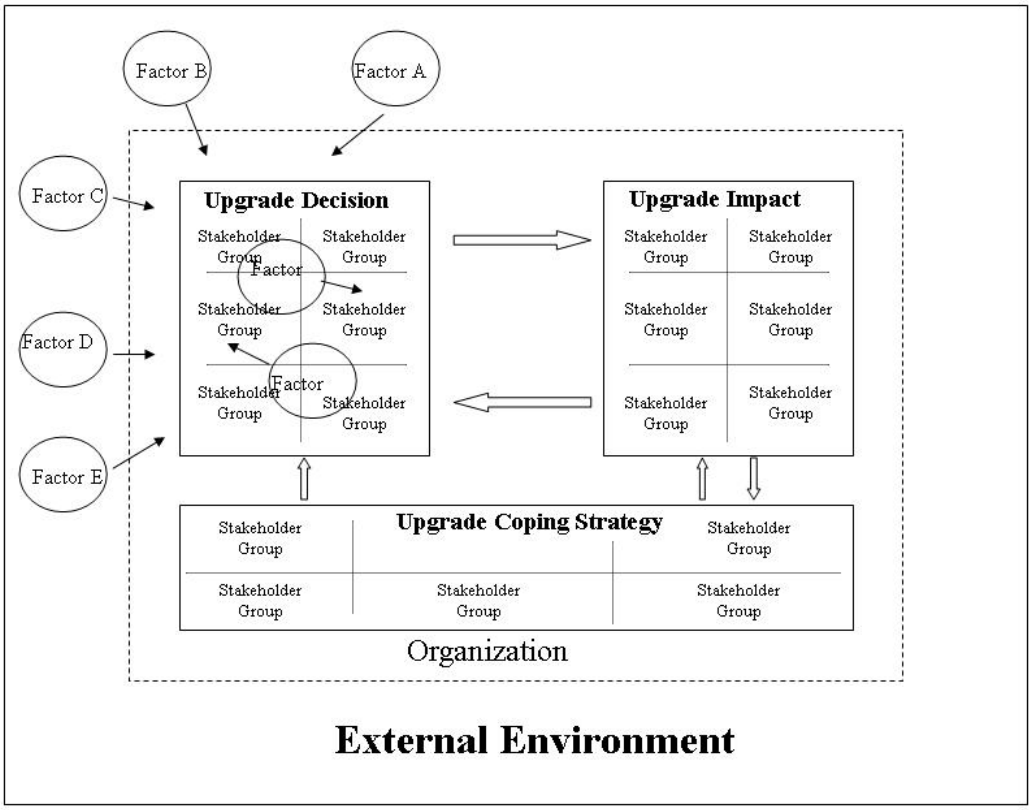

Figure 1 Conceptual Framework

4.3.1.1 Description of the Diagram

There are three boxes in Figure 1, Upgrade Decision, Upgrade Impact, and Coping Strategies. They represent the three main areas that this research explored. Each box is subdivided by a dotted line into sub boxes that represent different stakeholder groups. This is to show that the phenomena of that particular research area were studied from the perspectives of multiple stakeholders. The organization boundary is drawn using a dotted square box with External Environment surrounding it. The purpose of using a dotted boundary is to show that environmental forces can permeate the organization and create influences on various decisions or policies that the organization makes. This is to capture the interaction between the organization

and its external environment. In addition, the circle in external environment represents different environmental forces that could influence the organization. At the same moment, there are forces within the organization that could also have influenced the decision to upgrade. Internal and external forces only pertain to upgrade decisions.

The second research question, which looks at perceptions of upgrade impacts, is represented by the box labeled Upgrade Impact. The third research question, which explores how an organization copes with the effects of upgrade, is being explored under the box Coping Strategies. Even though there are arrows between boxes, the framework does not suggest a causal relationship between Upgrade Decisions and Upgrade Impact. The arrows are placed to show the possible sequence of stages in an upgrade process, i.e., a process model rather than a causal model.

The above framework was used to provide an initial guide on how the study could be conducted and did not make the study less exploratory or inductive. Based on past literature and pre-conceptions that underlied this research, a conceptual framework served as a starting point to the study. As Miles and Huberman (1994) pointed out, a conceptual framework "can be changed en route." There is no doubt that the conceptual framework evolved as the research progressed and new insights were gained. Modification to research design is inevitable, especially in exploratory research.

4.3.2 Case Study

"A case study is an exploration of a 'bounded system' or a case over time through detailed, in-depth data collection involving multiple sources of information rich in context" (Creswell 1997). It is especially suitable for studying contemporary phenomena in a situation

that the researcher has no control over, and answering questions that are related to *what*, *how* and *why* (Yin 1984).

The case study is commonly associated with exploratory research with a purpose of theory building. Eisenhardt (1989) developed a road map to build theories from case studies by integrating previous work on qualitative methods (Miles and Huberman 1994), design of case study (Yin 1984), and grounded theory (Glaser and Strauss 1967). Because the purpose of this study is to explore the issues of packaged software upgrades, which has not gained much attention and without much prior theory, the logical deduction approach to theory testing is not used here. In fact, a single site, multiple case study is used to explore the phenomena, build constructs, and discover potential relationships between constructs and their relationships. The goal is to build a substantive theory from data (Glaser and Strauss 1967).

A case is defined as an upgrade project in the organization. Bounding the case around an upgrade project allowed the study to focus on the details of a particular upgrade, and investigate the circumstances that influenced the decision, impacts associated with it, and also the coping strategies used in the project.

A single site, multiple case study is favored because different upgrade projects within an organization can be studied under a controlled organizational context, with focus on the decisions, impacts, and coping strategies used in different software upgrades. Having multiple cases inevitably dilutes the analysis of individual cases (Creswell 1997). Nevertheless, it allowed me to see how the theory "works out" (Miles and Huberman 1994) because similar phenomena were observed repeatedly in different technology upgrades. By studying multiple upgrade projects, I was able to discover commonalties of different projects, regardless of the technology, and also understand the differences between upgrade projects due to characteristics of

technology. It also allowed for a broader coverage of the issues and improved confidence in the findings.

In addition to the field study, two pilot studies were carried out in one of the colleges at a state educational institution. Approximately 18 subjects from three stakeholders group were interviewed: management, IS professionals, and users. The purpose of the pilot study was to fine tune interview questions to guide data collection in the main case study site.

4.3.3 Site Selection

One of the most important steps in case study research is to select the case(s) that provide the opportunity to study the intended phenomena. There are several different strategies in selecting cases. A maximum variation or polar case allows for comparisons of extreme cases whereas comprehensive sampling tries to examine every available case, which can be time consuming and impractical (Creswell 1997). Creswell advised researchers to choose a case that is "most promising and useful." In general, case study methodology involves purposeful sampling that lets researchers select the case based on the intention of the study (Miles and Huberman 1994; Creswell 1997).

The ideal site is a large organization that has multiple stakeholders, employs different types of software packages, and has either conducted package software upgrades in the recent past or has ongoing software upgrades.

The site selected is a *Fortune* 500 company, with its headquarters located in the South Eastern part of the United States. The company is referred to as Southeastern Company (SE Co.) throughout this study to conceal the identity of the organization. SE Co. employs 75,000 employees worldwide and is made up of two large divisions: one focus on manufacturing consumer products, another focus on developing construction materials.

Between the two divisions, the Consumer Product division (CP) is the star performer and accounted for over $3 billion in sales for the years 1999-2001. The division has grown tremendously through aggressive acquisitions, and it had approximately 120 manufacturing facilities throughout North America at the time of the study. When the study began in April 2002, SE Co. was in the midst of splitting the two divisions into two independent organizations, but the plan was deferred due to economic downturn. I was placed in CP division because of my intention to study major technology changes like ERP upgrade. The director of SAP group, Angela, was assigned as my field contact by the Chief Technology Officer.

Prior to entering the field, I planned to conduct two types of case studies: one based on retrospective accounts of past upgrade projects in the organization and one focused on an ongoing upgrade project, as in Leonard-Barton's (1990) dual method. However, when I arrived at the site of study, there were no ongoing upgrade projects, and the decision was made to conduct both cases retrospectively.

Two major software upgrade projects at CP were selected: an ERP systems upgrade from SAP 3.0F to 4.6C and an operating systems upgrade from Windows NT4.0 to Windows 2000. Windows 2000 upgrade was a good candidate to study because it was a major upgrade that spanned the entire CP division, and it had just been completed a month prior to the study. The SAP 4.6 C upgrade was also a good candidate for the case study because it was a major upgrade that greatly affected both IT personnel and users. The SAP upgrade was the last upgrade project on SAP systems and was conducted in December of 2000. Even though 15 months had lapsed between the upgrade and the beginning of the study, most stakeholders remembered the case clearly.

4.4 Data Collection

Multiple data collection methods are recommended in conducting a case study. "Accordingly, qualitative researchers deploy a wide range of interconnected methods, hoping always to get a better fix on the subject matter on hand" (Denzin and Lincoln 1998). Yin (1984) suggested six sources of evidence: documentation, archival records, interviews, direct observations, participant observations, and physical artifacts.

Of these six data sources, interviews are considered as one of the most important sources of case study information. Yin (1984) identified three types of interview: open-ended interview, focused interview, and structured interview. Open-ended interviews allow respondents to speak freely on a topic. In contrast, focused interviews are usually short with interviewers asking questions from a pre-planned protocol. This is similar to what is commonly known as a semi-structured interview (Miles and Huberman 1994). Finally, a structured interview is similar to a formal survey because specific questions are used to interview participants.

For this study, interviews with various stakeholders were the main source of data because stakeholders' perceptions on how technology upgrades affected their work was the key question. Both open-ended and semi-structured interviews were used to discover phenomena, and a list of interview questions was used to guide the process. Section 4.4.1 shows the interview protocol that was used as a preliminary guide to interview.

A snowball sampling technique was used to solicit interview subjects. Through my contact, Angela, I was given access to four departments: Infrastructure Support Department, Database group, UNIX Operating Systems group, and SAP Support and Development Department. Angela would first put me in touch with the senior manager of each department and I set up my first interview with them. After speaking to the manager in charge of the department,

I was usually referred to the key person who was responsible for the particular project, and through the key person, I was referred to other members of the group.

Two stages of data collection were conducted. One began in April 2002 and lasted nine months. During that period, both first-time and follow-up interviews were conducted. A second data collection stage was conducted in May 2004 for two days. Over the course of the research, 57 subjects were interviewed generating a total of 67 interviews, each lasting between 60 and 90 minutes. Of those interviews, seven were phone interviews of people at two manufacturing facilities. Thirty-three people were approached for interview of the Windows 2000 project, twenty people for the SAP 4.6C, and four people in SAP-affiliated department for context information. However, users who experienced both systems were asked about both systems. All interviews were tape-recorded using two tape recorders for fear of losing data; in two cases, the back-up recorder strategy proved to be life saving. All phone interviews were conducted at home using a speakerphone to allow for tape recording.

Direct observations were also conducted to improve my understanding of the work context of the study. Some of the activities that I observed were: a data center tour, a interdepartmental meeting between the SAP group and its affiliates, user training sessions, and observation of order entry on SAP. I also had opportunities to sit in technology governance committees meetings to observe the decision process where discussions on the Windows XP upgrade were held. Even though some of these activities were meant for upcoming projects, the opportunities to observe and participate allowed me to relate better to stakeholders' accounts in the two cases that I conducted. In addition, documentation to supplement interview data was also collected from interviewees, SE Co.'s intranet, and the Internet.

4.4.1 Interview Protocol

During the interview, I introduced myself, stated the purpose of my study, made the subject feel at ease, and proceeded with the questions listed below. The list is an inventory of interview questions sorted according to the categories that they belong to. Not every question listed was asked during each interview. Due to the inductive nature of this study, questions were chosen based on subject's area of expertise and answers provided.

List of background information that was collected on the organization

Industry

Geographic location

Type of business

Organizational structure

IT departmental structure

Decision structure for IT adoption (Who makes the decision?)

Size of the user population served by IS department.

Type of technologies that are being discussed

List of background information that was collected from interviewees

Name

Position / Rank in the organization

Type of stakeholder (user, management, IS professional)

Years with organization

Education

Past working experience

Years using technology

Interview questions on general background of upgrade issues

How often is technology upgraded in your organization?

How does your organization keep up with the new technology?

Do you conduct "environmental scanning"? Which type?

Does your organization have any policies (formal or informal) on software upgrade?

If YES

What are those? (Or, can you tell me more about them?)

Who makes those policies?

If NO

How do you decide if the software needs to be upgraded?

Who makes the decision to upgrade or not to upgrade?

What type of upgrade strategies would you say your organization adopts?

General or specific? Proactive or reactive?

In general, how is upgrade being executed?

Organization-wide (Usually executed by system administration group responsible for networking, OS, end user application etc.)

Departmental wide upgrade (Special software used by certain department)

Who implement the upgrade?

Individual upgrade (Is there a case in which users can buy their software through superior approval and then upgrade their own system?)

Interview questions on upgrade decisions

What influence the decision to upgrade or not to ugprade a specific software?

a. Window XP is out; any plan to upgrade? Why or why not?
b. Who has input to the decision to upgrade?
c. Who makes the ultimate decision to upgrade?
d. What are the three top influences to upgrade?
e. If not upgrade? What is the reason for that?

New version of software emerges, promising new opportunities. Do you embrace each new version? How do you know if it is right for you? When do you know it is time to upgrade?

When a new technology is released, how do you decide to upgrade or not to upgrade?

Are there *several common* factors that affect upgrade decision?

What is *the* driving force to upgrade this particular software?

Are there any external forces (government, vendor, partner, customer, and competitor) that influence your decision to upgrade? How?

Do internal users have any input to upgrade? Or, who has input to upgrade decision? Or, did you have input to this upgrade project? If yes, what was your input?

How do users influence your decisions?

Is your company doing any kind of software upgrade currently?

If YES - What is it?

If NO - do you plan to conduct any in near future? OR / AND

What is the last upgrade project? Which type of technology?

I understand you are upgrading _____

Why is this type especially important?

Interview questions on impacts of upgrade

General

How do you see upgrade in general?

What is the trade off for keeping old version of software vs. upgrading to new version of software?

How far behind is "ok"? How long can you survive on not upgrading? What is the catch of continuing to use current software?

In the past, were there any cases in which the company would have been better off to upgrade (or not upgrade) the software? What happen?
What are the impacts?

Does it always pay off to upgrade to new technology?

Can upgrade create problem for organization?

Have you ever skipped a version of upgrade? Did you have problem with the upgrade later?

When is new technology beneficial?

Does your organization keep track of the improvement brought by upgrade?

Do you have any formal measures of upgrade improvements?

How do you measure cost and benefit of upgrade?

Positive Effects

What are some of the expected improvements when conducting an upgrade?

What are some of the most common performance improvements resulted from upgrade?

How do you gauge improvements resulted from upgrade?

What is the best upgrade experience that you have? Why?

Negative Effects

Have you had any problems during upgrade?

How long did it take to fix the problem?

Is having problem during upgrade normal?

How did the problems affect your productivity?

Why conduct upgrade if it gives you problem?

What are the most common problems encountered during upgrade of software?

Does the type of technology create different types of problems?

Coping Strategies

How do you feel about constantly receiving service packs or sub-version upgrades sent by software companies?

How do you cope with the situation?

Do you conduct every upgrade that you receive? Why or Why not?

What type of strategies do you use to cope with those frequent upgrades?

What do you usually do when upgrade creates a problem?
How do you cope with it?

Does the organization use any strategies to cope with constant upgrade?

What are those?

How do you evaluate the effectiveness of each coping strategy?

Of all coping strategies that you have known, which is deemed the most effective? Why?

4.5 Data Analysis

The data analysis process began as the data were collected. Unlike quantitative data analysis, which usually comes after all data have been collected, interweaving the two stages of data collection and data analysis is encouraged in the qualitative approach. It allowed me to revise the data collection strategy based on what I had collected so far (Eisenhardt 1989; Miles and Huberman 1994).

According to Eisenhardt (1989), "A key feature of theory-building case research is the freedom to make adjustments during the data collection process." Adjustments are necessary, especially in exploratory, theory-building research because the research is often loosely defined to discover interesting new findings from the field. Some of the adjustments that can be made as the research progresses are: addition of cases to investigate emergent themes; modification and addition of questions to the interview protocol; and addition of new data sources as necessary. Even though flexibility is allowed, it should not be viewed as "a license to be unsystematic" (Eisenhardt 1989). It should only be conducted with caution and applied where it is truly needed.

In general, data analysis is considered the most difficult part of case studies due to the lack of codification (Eisenhardt 1989; Yin 1984). As Yin (1984) put it, "The analysis of case study evidence is one of the least developed and most difficult aspects of doing case studies."

Eisenhardt (1989) also supported this view by saying, "Analyzing data is the heart of building theory from case studies, but it is both the most difficult and the least codified part of the process." One approach to counter this difficulty is to keep in mind the ultimate goal of the study while conducting the analysis. According to Yin (1984), the goal is "to treat the evidence fairly, to produce compelling analytic conclusions, and to rule out alternative interpretations" (Yin 1984). A high level strategy that I adopted was to conduct both within-case analysis and cross-case analysis. Nevertheless, before any single case analysis can be conducted, data collected needs to go through a detailed analysis called coding.

4.5.1 Coding

According to Miles and Huberman (1994), "Coding is analysis." It is a process that breaks down data, conceptualizes them, and puts them back to discover concepts embedded within the data. Strauss and Corbin (1990) suggest three types of coding: open coding, axial coding, and selective coding. Each of these coding types creates a more general grouping than the previous one. Likewise, Miles and Huberman (1994) described three types of codes that can be used to tag meanings to a "chunk" of data, be it words, phrases, sentences, or paragraphs. Those codes are descriptive codes, interpretive codes, and pattern codes.

Descriptive codes are used to assign "a class of phenomena to a segment of text" (Miles and Huberman 1994). They belong to the first level of analysis in which data are summarized into segments. At this stage, a low level code is assigned to all segments of text that fall under that class of phenomena, and not much interpretation is done. Later, if the same segment of text assigned to a descriptive code is analyzed with more interpretation, an interpretive code is employed.

A pattern code is considered a second-level analysis code, which contains more explanatory power and is more inferential. It is used to group those segments in first level analysis into higher level themes or constructs. According to Miles and Huberman (1994), pattern coding has four important functions:

1. It reduces large amounts of data into a smaller number of analytic units.
2. It gets the researcher into analysis during data collection, so that later fieldwork can be more focused.
3. It helps the researcher elaborate a cognitive map, an evolving, more integrated schema for understanding local incidents and interactions.
4. For multi-case studies, it lays the groundwork for cross-case analysis by surfacing common themes and directional processes.

Three rounds of coding were conducted. Initial coding was done on an electronic copy of the transcript with descriptive codes assigned to small sections of text. On the second round of coding, descriptive codes and the corresponding texts were copied into a word document for further interpretation, and similar themes were grouped into named categories. At this point, groupings began to appear. Six documents were created to correspond with the three research questions for each technology: decisions for OS, coping strategies for OS, impacts for OS, decisions for SAP, coping strategies for SAP, and impacts for SAP. For each of these documents, a third round of data analysis was conducted to derive higher-level concepts and specific lower level concepts within them. An example of the groupings for the SAP decision document is shown in Figure 2. Each of the entries in Figure 2 was linked to a later section of the document containing quotes from the transcripts. Figure 3 displays a section of quotes linked to the "Functionality" entry in Figure 2 Table of Contents. The documents were used to assist within-

case analysis and the write up of both cases. To perform cross-case analysis, tables were created to compare and contrast similarities and differences found in both studies. For this study, data analysis continued through the write up.

Because the research is exploratory, no theory was selected before the study began. To induce the model, a second round of literature search was launched to look for theories that could help explain and confirm some of the phenomena observed in the cases. The resultant model is presented in Chapter 7.

An Example of Table of Contents for SAP Decision Document

……..

General Policy on Upgrade 8
Company Policy to be on Support 8
Try to Do One Every 18 months 8
Do not want beta version 9
General Driving Force to Upgrade for IT 9
Context (Don't Do Cost and Benefit Analysis) 9
Internal Business / Business Needs 9
Context (Business Does Not Like To Upgrade in General) 10
Business Needs Can Initiate Effort to Look into Potential Upgrade 10
Motivation to IR (In Terms of Benefit Derived from Software) 10
Replace Custom Code 10
Influences of Decision to 4.6C Upgrade 11
IT Member Not Informed 11
Timeline - Acquisition of FJ 11
A different view 11
Two Main Reasons - Support and Functionality 12
Functionality 12
Context - Importance of SAP Support 14
Support 15
IT perceived fairness of Vendor stop support 17

.....

Figure 2 Partial Table of Contents from the SAP Decision Document

An Example of Content for Functionality entry in Table of Contents for SAP Decision Document in Figure 2.

……..

Functionality

[Decision (SAP 46C): Functionality needed] So, a lot of different reasons but the bottom line is we need to upgrade because we needed a lot of functionality. We need a lot more functionality, a lot more flexibility with the system, there was a modest fixes that we applied that now were part of the new upgraded version. – Rose (2)

[Decision (46C): Functionality in new release] It's was working ok for us and then we started to acquire more businesses and we were growing, the flexibility that we want to have on our software were not there in the older releases. The newer releases were so much better than the old one so -- Angela (9)

[Decision (46C): Functionality] So, SAP was starting to deliver more functionality and the only way to get it was to get on a newer version of SAP. So one reason was for bugs, the other reason again was functionality.
The business PAUSE wanted to do some stuff or said, "man I wish the system could sort a certain way." And, then we'd write SAP and say "…" and they'd say "oh yeah, lots of customers have requested it and that's in the next release." Or, "I wish you had a report that did X, Y, Z" and before we develop one, cause we'd say "well that seems something basic, why wouldn't SAP just develop that for everybody who buys SAP?" And, then they'll say "oh yeah, we did do it. But, you got to buy release 4.6 and you'll get to it or um…." You know that SAP is really hard to navigate in this screen. We want, our users' saying "we like SAP but it needs to be more user friendly and web interactive." "Oh you can if you upgrade to release 4.6." So, it was, we needed to because SAP was really improving their functionality. -- Tina (5)

…….....

Figure 3 Excerpts from Decision Document

4.6 Interpretive Case Study Evaluation

Since this is an interpretive case study with goals to explore the upgrade phenomenon and potentially induce theory, seven principles of interpretive research proposed by Klein and Myers

(1999) were used as the primary guide to evaluate and guide this research. This section lists each principle along with the steps that were taken to observe each one.

The first principle is the *hermeneutic circle,* which influences all other principles. Basically, it says, "All human understanding is achieved by iterating between considering the interdependent meaning of parts and the whole that they form." Because the research objective is to examine an organization's decision, impacts, and coping strategies as a whole, yet understand each area through individual stakeholder, the first principle served as the overall guideline to the study.

Second principle is *contextualization* of the phenomena. Researchers should base their research in the social and historical background where the subject matter is and allow readers to see how the investigated phenomena emerged from those contexts. This principle was observed by conducting case studies in their natural settings and by paying close attention to contextual information when collecting data. Furthermore, I carefully analyzed the case in reference to its social and historical environment when conducting data analysis.

The third principle emphasizes *interaction between researchers and the subjects*. Following this principle, the researcher needs to recognize that the data collected in the study are the product of social interaction between researchers and research participants. To follow this principle, I was conscious when conducting interview because I understood my presence and my reaction to interviewees' response could affect the answers that I received. I took steps to minimize my influence in data collection wherever possible and explicitly acknowledged my influence where it was noted.

The fourth principle is the *abstraction and generalization*. This principle is used to explain how "theoretical abstractions and generalizations should be carefully related to the field

study details," and to allow readers to understand how the findings are reached. This is an important principle that I followed by providing sufficient details to readers and by presenting a logical argument that tied the details and findings together.

The *principle of dialogical reasoning* is the fifth criterion listed. It is meant to raise awareness that researchers' theoretical preconceptions may not always match with the actual findings. To apply this principle, I derived research findings from data and used past literature as a guide for potential explanation. I also remembered not to force data into any prior theory. Instead, I carefully considered the relevance of the data without any preconceived explanation before making conclusions.

The sixth principle is about having *sensitivity to multiple interpretations* that can exist among different participants. This principle is important because different participants can have different perspectives on organizational issues. This study adopted a stakeholder analysis to allow different interpretations to be carefully examined and to understand the meaning that they carried.

The last principle reminds us that we need to be *suspicious* of data collected from participants. We should not just take their words at face value. For this principle, the explanations given by stakeholders were interpreted and corroborated with other sources of data.

CHAPTER 5

CASE NUMBER ONE: WINDOWS 2000 UPGRADE AT CP

Windows 2000 upgrade was a huge project that had taken over a year to implement from divisional (CP) headquarters to all location. The implementation of Windows 2000 was carried out by the Infrastructure Support Department (ISD), which is led by a senior manager, Sonny. The Infrastructure Support Department has over 60 IT employees, spreading across four geographical locations that report directly to the senior manager. They are responsible for the desktops, laptops, servers, and network connectivity at the main headquarter location and three smaller sub-headquarters locations in other parts of the country. In addition to the four headquarters locations, the senior manager also has “dotted line” authority over all manufacturing facilities and warehouses. However, the management of day-to-day IT operations and support was described by the senior manager to be “very decentralized” within CP. While divisional headquarters set strategic direction, and managed the operations of four headquarters locations, each manufacturing facility had its own IT department.

> *For something we do, for the day-to-day operations or supports, we have a very decentralized model. So if you take any particular type of [manufacturing facilities] out in the field, they have a local IT support staff, a local IT manager. And so they are directly responsible to the [manufacturing facilities] for the operation and making [product] and how computing and all software helps make that [product], helps the business. I do have some responsibilities for those folks though. If they have issues or problems, they call my group. And I'll try to help, straighten it out by either fixing*

ourselves or getting the right people from the corporate groups involves to solve whatever the problem is. – Sonny, Senior Manager of ISD

The deployment of Windows 2000 was a concerted effort among different job titles within Infrastructure Support Department (ISD). ISD is composed of two large groups: (1) Operations Support group and (2) Projects and Design group. Operations Support consisted of Desk Side Support personnel and Servers Administration personnel. They managed, ran and supported the day-to-day desktop and server activity, whereas the Projects and Design group focused on special, one-time IT projects. Within the Operations Support group, Desk Side Support personnel were also the first line of defense for any help desk calls that came in. They were responsible for all computer users at the division headquarters, and each person was assigned one or more floors of computer users outside of their department. Server Administration personnel were a group of four to five highly technical IT people whose focus was more on the backend server and networking. In addition, they provided technical support for any issue that Desk Side Support personnel could not resolve.

Even though CP was an autonomous unit in IT decision making, it relied on the corporate IT unit, Computer Shared Services (CSS) for research, evaluation and recommendation of "common" technologies, like Windows operating systems and Office applications, which were used by all divisions.

We have certain groups, the corporate groups not in corporate division but overall in corporate upstream… that are essentially dedicated to trying out the new hardware and looking at new software, new solutions. That's how we keep abreast of things. We trust those

groups to tell us what's realistic. Do we really need to go to new hardware? Do we find ourselves on the latest and greatest, do we find ourselves one step behind?

– Sonny, Senior Manager of ISD

5.1 Corporate Governance Process

Before any commonly used software product could be adopted corporate wide, it had to be endorsed by two steering committees through a technology evaluation process that made the product a corporate standard. Both of the steering committees, the Technological Architecture Committee (TAC) and the Technological Steering Committee (TSC), were headed by Victor, the director of corporate IT Strategic and Planning department. TAC and TSC were comprised of representatives from each division, and they meet once a month. One of the tasks of these committees was to set corporate standards for new technology or product upgrades.

Before a technology could be proposed as a corporate standard at TAC meeting, the corporate group in charge of that technology would have to evaluate the technology first. Windows operating systems was a common technology governed by NT Platform Solutions (NTPS), one of the Corporate Shared Services. NTPS was responsible for keeping up with the latest version of OS, following its new development, and determining whether to bring in the new version for initial evaluation. Preliminary evaluation usually consisted of a “proof of concept” type evaluation, a paper evaluation, or limited systems evaluations to determine whether the new technology was worthy of further pursuit. If the technology was deemed useful for the company, then more elaborate testing was coordinated within the group that was responsible for the particular technology, and within other Corporate Shared Service groups to begin testing for compatibility with interdependent applications. At this stage, any issues with

new technology were noted, and solutions were sought to make it workable with existing applications. After the new technology had been examined to learn about its features and fit with current software and hardware, representative from NTPS put in a request to present its findings to steering committees and proposed to make the technology a corporate standard.

Sometimes, an ad hoc committee was formed to evaluate the technology before it was presented to TAC. The committee was called Technology Evaluation Committee and was comprised of people who were familiar with the technology. Unlike TAC or TSC, the members in TEC were formed on an as-needed basis.

Once enough testing was made, the representative from NTPS or TET gave a presentation to the TAC committee. The members of TAC were representative of senior technical people from each division within the entire corporation. Based on the presentation, the TAC members voted on whether to make a proposed new technology a corporate standard from a technical standpoint. If doubts were raised in any circumstances, TAC requested NTPS or TEC representatives to come back and provide more information in the next meeting.

After a technology is voted a corporate standard in the TAC meeting, the decision was brought to TSC, which was comprised of senior management of IT. TSC meetings usually followed TAC meetings. In their meeting, TSC members weighed the financial and strategic aspects of company direction and made the final decision to accept or reject the new technology as a corporate standard.

Windows 2000 was first brought to the attention of TAC in January of 2000. It was more of a report to TAC about Active Directory, a key technology that promised to bring tremendous improvements to backend operations. Following the initial report, several presentations were made to TAC on the progress of Windows 2000 evaluations. A Technology Evaluation

Committee (TEC) with representatives from all divisions within the organization was also formed to discuss the network structure best suited to implement Active Directory. The director of the group responsible for evaluating Windows 2000 explained that the process took longer compared to many other IT projects because the technology was completely new to them. A Microsoft consultant was also brought in to assist with the design and implementation issues. One year after the first presentation at TAC, Windows 2000 was recommended for adoption as a corporate standard.

Even though a division could adopt a technology before it became a standard, the practice was not advisable and was seldom carried out. One of the reasons was that a non-standard product that had not been tested for its compatibility with other related applications would not receive any support from Corporate Shared Services. Zack, a manager in the ISD group provided an example.

> *Windows XP is already out on the market and a lot of people at home are using it but we haven't started here yet. We're still looking at it, we're testing it and we were told not to proceed with it until we're ready, as far as a corporate standard. So everything is thoroughly tested before we move to it. And, there are groups here within CP that all they do are they test new products.*

5.2 General Policy on Managing Technology Changes

IT management did not consider SE Co. as a "technology company," and the strategy that they adopted was not to be "too far in front of technology." As a senior analyst at NTPS, Frank,

pointed out, “We don’t try to keep at a bleeding edge. We’re more of a cutting edge where anything new comes out. We’re looking at it but we’re not immediately going to it.”

Typically, there was a waiting period for the vendors to work out the “kinks and bugs” and let new technology matured before IT personnel from corporate CSS would even look at it. In fact, the policy was not to evaluate any technology until the first service pack had come out. According to Gary, a senior manager in CSS, this gave them two advantages:

> *… hopefully all the issues will work out before we get there but also the [company name], they’re kind of more in front than us, we can leverage other people and other experiences in the industry to see what issues have they run into, what challenges have they run into, how does it integrate with their current environment. So, we get great advantage from doing that.*

Even though the policy did not encourage immediate adoption of any new technology, CSS always kept “an eye on industry trends” by sending IT personnel to conferences, having employees read trade magazines, speaking to companies of their same size in the region, and also subscribing to professional consulting services like Gartner Group. In addition, they were in constant contact with vendors for information on new product development.

5.3 Influences on Operating Systems Upgrade

When it comes to operating systems upgrade, a senior manager explained, in many cases, his group did not upgrade for cost savings, but for “things that were broken in the previous version.”

One factor that potentially affected the planning for upgrades was the vendor's product sunset date. As in most packaged software, vendors continue to release patches for their products as new bugs are found or security breaches occurred. In the case of Microsoft products, the company releases "service packs," as often as every few weeks, that can be applied to current operating systems. After a version of operating systems has been released for several years, Microsoft gradually phases out its support in hope that the customers will move to the next version. The date where no more support will be provided is called the "sunset date." As a large organization that uses mainly Microsoft products, SE Co. follows the "sunset dates" closely. There are two sets of sunset date:

> *They have what they call the mainstream support and then they have the extended support date. Generally that's a year between the two. The mainstream support is basically, "We're not going to create any more service packs for this operating system, we're not going to create any more hot fixes." If there's a need for a security hot fix, they will come up and fix that, during that timeframe.*
>
> *Now the extended support date is the drop dead date. At this date …no service packs, no updates, no new security packs, no nothing. If you want support it's going to cost you $100,000 a year to support it plus, if they have to create a hot fix, then it's $375 an hour for each development hour that it takes for them to develop it.*
>
> – Frank, Senior Analyst of NTPS

Vendor support was critical to SE Co. because they relied on vendor service packs for security or bug fixes. Thus, a vendor's sunset date was very influential to IT planning. As a senior analyst put it, "Really, the whole driver of it is around Microsoft sunset." Ideally, when it came to managing operating systems upgrade, CSS would have liked to "ride it out as long as possible." According to the director of the Strategic and Planning Department, the question was "at what point do you have to get off Windows NT, so how long can it last?"

5.4 Influences on IT's Decision to Set Windows 2000 as a Corporate Standard

Nevertheless, Windows 2000 upgrade was a different case. The vendor's sunset date was not the driving force, as the senior manager of CSS, Gary, explained:

> *… it wasn't necessarily a sunset date driving us toward this [Windows 2000]. We didn't sunset NT4 or Windows 95 or any of the other prior operating systems at that point. In fact we'd been doing Windows 2000 for a while. Now, we're just getting around to sunsetting Windows 95 and NT4. …*
>
> *In the Windows 2000, part of the big push for Windows 2000 was the Active Directory, and what the Active Directory brought to the equation was the fact that it would reduce your desktop management and it would help with application deployment. It had self-healing properties; it had management capabilities; and things like that.*

What attracted corporate IT to Windows 2000 was a new technology component called Active Directory. Active Directory was a directory service included as part of Windows 2000. It was used mainly to manage server and networking objects. Even though Windows 2000 could be deployed without implementing Active Directory, an organization would not fully capitalize on the true benefits of Windows 2000 technology without it. Active Directory allowed system people to manage distributed computing environments centrally. This was an extremely attractive feature to SE Co., which had over 25,000 desktops throughout 700 different locations worldwide. In the past, distributing software to users had been a nightmare; two not very efficient approaches were used by IT department.

> *In order to deploy software we have to go either pushing out using SMS - an application of Microsoft - which is very flaky and it's not widely distributed. It's not very reliable and requires desk side visits, which is very time consuming, resource intensive. Everybody, the TAs had to go and visit each of the desktops, install the apps. What Active Directory allowed you to do is push applications and standards to the desktops remotely*
>
> – Frank, senior analyst of NTPS

Thus, having the capability to distribute software remotely in a reliable manner was an immense breakthrough to IT people who had been visiting users with a disk in hand. Unlike its predecessors, in which improvements were incremental, Windows 2000 was a leap of technology over previous versions of the operating system. It was considered a new technology that SE Co. could greatly benefit from. Windows 2000 became a corporate standard in January 2001.

5.5 CP's Philosophy in Managing Technology Changes

At the CP division, new technology was carefully managed to strike a balance between the necessity to keep up with technology and the costs to deploy the technology. Sonny explained the guiding philosophy that he used to manage technology changes in his division.

> *My goal is to provide a computing environment that never is a stumbling block for the business. The business comes back to us and says, I want to do this, I want to do that, whatever that is, a new software package, or some e-commerce initiatives or something on those lines. Either customers are demanding or internal business customers are demanding. So my goal is to make sure the computing environment never gets in the way. If they come to me, we already got the infrastructure in place to do whatever they want to do. … I don't want to be on the bleeding edge, I don't necessary want to be on the leading edge, but I want to be close so I have the minimal amount of effort if the business asks me to do something that's on the leading edge.*
>
> – Sonny, Senior Manager of ISD

He went on to explain they did not conduct an upgrade on every new version of operating systems that was released to the market. Although he did not manage on the basis of minimizing the cost of IT, a strong business case was required for an IT project to gain approval.

5.6 CP's Decision to Upgrade to Windows 2000

Having representatives in both technology steering committees and the Active Directory design team, Sonny was well aware of the technical features in Windows 2000 and the potential improvements that Active Directory could bring to software deployment and remote desktop management. Shortly after Windows 2000 became a corporate standard, Sonny made a decision to migrate to the new operating system. At CP, the power of decision within a department usually rested upon the department head; none of the users or IT personnel interviewed were fully aware of the defining factors behind the decision. Although he had to check with upper IT management that oversaw all IT departments, he was usually the decision maker.

What triggered the decision to upgrade was a chain of events that began with an $8 billion acquisition of a company twice the division's size. The company is referred to here as Newly Acquired Company (NAC). The acquisition created two groups with very diverse information systems. NAC used "homegrown" mainframe applications for order management and SAP for accounting whereas CP used mainly SAP for business transactions. Following the acquisition, the decision was made to integrate business systems from the two companies. As a result, CP's SAP system was chosen to replace NAC's mainframe applications.

Before the system integration project could march forward, the IT infrastructure group determined that it was necessary to replace the hardware at NAC. Because users at NAC mainly used terminal emulation programs to access the mainframe, they did not need powerful desktop computers. Compared to CP, most NAC users were on older microcomputers with 32 MB RAM whereas users at CP used newer machine that has at least 128 MB RAM. The decision to upgrade NAC's hardware opened the door to standardize all operating systems within CP.

Following the acquisition, three platforms of operating systems co-existed between two companies: Windows 95 for NAC, and both Windows 98 and Windows NT 4.0 for CP. According to Luke, project leader for Windows 2000, a company could not have standardized applications unless it also had standardized operating systems. NAC's hardware upgrade gave ISD an opportunity to move all users to a single operating system.

> *So we want to move everything to Windows 2000 because we've been testing, we've been participating in the Active Directory pilot. So we knew the value of deploying software, fixes, and software packages with the group policies, part of Active Directory. We knew the reduction of administrative cost with Active Directory.*
>
> – Sonny, Senior Manager of ISD

Even though Windows 2000 was considered a paradigm shift in backend system administration and its benefits were phenomenal, without the need to replace hardware for NAC, CP would probably have waited longer before making the move. According to Sonny, "Right now, we are complete with the upgrade of Windows 2000. I think if we didn't have [NAC], we wouldn't be completed." The need to upgrade more than 3,000 desktops for the system integration project opened the door for a large scale IT initiative.

Essentially, according to Sonny, the decision to upgrade was in large part influenced by two factors: the benefits of Windows 2000 and the need to lay down an infrastructure for NAC.

Because if you look at the two arms of the business case, one was we are replacing a bunch of hardware in [NAC]. Number two was there is awful lot of support, software rollout support benefit to go on Windows 2000.

At CP, IT projects required intense resources and were justifiable only if real business needs were present. Although CP by itself could have stayed on the previous operating systems, the need to replace NAC hardware for the system integration project, an opportunity to standardize all users, and the attractive potential of Windows 2000 all combined to make a strong argument for the deployment of a division-wide upgrade.

5.7 Deployment

The Windows 2000 upgrade was a challenging project in many respects. First, Active Directory was a shift of paradigm in Windows technology and unfamiliar to most IT personnel. Second, the Windows 2000 upgrade project involved between 120 to 130 locations and approximately 10,000 desktop computers throughout the country; the project was not only massive, but geographically dispersed. Third, manufacturing facilities and distribution centers were products of multiple acquisitions. Each remote center managed its own IT activities, and seldom interacted with divisional headquarters. For the first time in CP's history, a standardized operating system would be deployed to every site.

The task to upgrade CP was a huge undertaking and the senior manager of Infrastructure Support Department wanted Luke, a cheerful, enthusiastic, highly technical, and extremely qualified IT professional from corporate IT group to lead the project. Luke was recruited from corporate IT and became a member of the special project and design team. He was a hands-on

project leader and heavily involved in every aspect of upgrade. During the nine months of deployment, he led a small team of people who traveled to manufacturing facilities and distribution centers, personally deploying Windows 2000 at almost every site.

5.7.1 Windows 2000 Upgrade at Divisional Headquarters

At CP, technical preparation for the Windows upgrade began months earlier. One of the first things that the Infrastructure Support Department (ISD) did was software testing. Although Windows 2000 had become a corporate standard, and has been tested for compatibility with other standard applications, ISD was responsible for making sure the deployment of Windows 2000 would not have an adverse effect on other non-standard software used in the division headquarters and manufacturing facilities. Before the upgrade could proceed, testing on all those software applications was essential. Different IT groups within the division began testing their applications on the new operating systems as early as six months before the official launch. Testing was an essential strategy used by ISD to safeguard against any technical issue. It eliminated the bugs and allowed IT people to seek solutions before the technology was put into use.

Meantime, the task of planning and coordinating the rollout of 3,000 users at divisional headquarters was delegated to Zack, a tall, 15-year veteran IT manager. One of the first things that he did upon receiving the assignment was get the user community to buy into the new technology.

> *Well, the way I sold it to them was, it was all enhancements for them, ok. Here is what I'm going to do for you. I'm going to give you a bigger, faster PC that will help your*

people be more productive. It will let them do some of their works that they do in half the amount of time since it is a faster PC.

However, not all users were excited about getting upgrade. For those users, Zack took a persuasive but firm approach to get his message across.

At the end, we had to tell them, well it's going to happen if you like it or not. This is the way the company is going. We are leaving an old technology and we're moving on to a newer technology, ok. So we try to be very upfront with them, explain all the pros and cons to it. Mostly pros, of course, and for those people that really didn't want to give in and they fought us, we gave them a date of when they need to conform.

One of the first things that Zack did to prepare for the upgrade was hardware planning. Windows 2000 raised the bar for hardware requirements. A microcomputer needs to have a minimum of 4 GB hard drive and 64 MB RAM to load Windows 2000. Even though many of the computers in use more than meet the minimum requirements, some were in need of memory upgrades. Each user's hardware was evaluated to determine whether he or she had the sufficient equipment to handle Windows 2000. Depending on the type of tasks a user conducts, he or she was classified as either a power user or a standard user, and hardware was assigned accordingly.

If that person was an accounting person that used multiple spreadsheets, 50,000 line spreadsheets and PowerPoint and Access and a lot of complications, I made sure that that individual got one of the better machines. Somebody in customer service that

basically uses a mainframe, … they really don't need that good a machine so I prioritized the user. …The power users are usually those who use applications other than the standard provided. These are people who create files larger than 1MB and use multiple applications compared to their peers. Thus, it is important that they get sufficient equipment to do their job.

-- Zack, IT manager

Prior to deploying the upgrade, Desk Side Support people visited each user to get a list of the software that he or she needed. In addition, users were required to sign a consent form agreeing to the upgrade. The manager explained, "This way they are completely in the loop, communications went out a couple of times …so there was no 'I didn't know.'" Communications were a big thing to the senior manager, Sonny, who believed in having good communication with users' communities to "keep them in the loop" of what's going on. "There's never too much communication," said Zack. At this stage of preparation, users' involvement was kept to minimum; IT personnel were the ones getting upgrade ready.

Official deployment of Windows 2000 was scheduled for summer of 2001. Nevertheless, ISD began loading Windows 2000 on computers prior to that. The policy was, if a user needed a replacement system or a new employee needed a new computer, he would receive the Windows 2000 operating system. This allowed ISD to move forward with new technology and reduced the need to put in Windows NT 4.0 and then upgrade to Windows 2000 a few months later.

A "build and swap" strategy was used to deploy the new operating system to individual users. Instead of visiting each user to install Windows 2000, IT personnel would come with a better computer pre-loaded with Windows 2000 for the user, and then swapped the old computer

for the new one. This approach reduced computer downtime at users' desks and allowed IT personnel to "refresh some of the hardware that was out there." All users received a better computer whether it was new or used. Through this approach, the computers switched out could be enhanced and given to other users.

The deployment was put under the supervision of Edward, the manager of the operations group. Several contractors were also hired for the project. To build these computers, a large cubicle was setup at the end of a hallway. There, new machines received Windows 2000 operating systems, and used machine went through memory upgrade and other hardware enhancements before contractors loaded the new operating systems. During the upgrade, Office 2000 was also put in to replace Office 97.

Even though most of the users had received the Windows 2000 upgrade before the study began, at CP, through my request, the operations manager arranged for me to observe a "build" process. James, a contractor who had been with the division from the early stage of the project, and Wayne, a summer intern, showed me how they upgraded a laptop for a new employee. The upgrade process was automated to reduce human interaction with the computer. James explained, "What we use is a two-disk standard install. There is a minimal amount of information that we have to put in, then just load everything from the server itself."

A Central Install Administration Team from corporate IT created the two-CD installation engine. It served as a "boot disk" and helped answer many basic questions prompted by the computer during installation process. During the main rollout, contractors were able to simultaneously upgrade eight to ten machines using the install engine. They began installation on a computer while software was loading the upgrade on another one. If a used computer was being upgraded, the CD would first reformat the hard drive, then partition the drive, and connect

the computer to the network and download Windows 2000. After the operating system was loaded, the contractor installed other standard software packages, e.g., Office 2000 and Norton Anti Virus. According to one contractor, James:

> *We get to the point where all was left for the Desk Side people do was to configure their Outlook for the end user, and then any custom application for any specific application, you know, they have project or Visio, something like that, that was specific to the users*

A typical installation took approximately 30 to 45 minutes to load the operating systems and an additional 15 to 20 minutes to load other software. Nevertheless, during the laptop upgrade that I observed, the upgrade took more than two hours due to several factors. This laptop installation was only Wayne's second upgrade, and he kept running into authentication problems that hindered the computer from connecting to the right domain on the network. The workaround was having IT personnel manually join the domain or using the administrator's id to login. In this instance, James logged in as an administrator and allowed the process to continue.

The network speed was also extremely slow during the installation due to server replication. Because all software was stored on the network, slow connection could affect the speed of installation. At remote sites, local servers were set up to improve the performance of download. In addition, Wayne and James had to install additional drivers and set up applications for remote connection, so laptop installations took slightly longer than desktop installations. After the computer was configured, it was stored in the inventory room.

At the division headquarters, the actual upgrade to individual users fell on the shoulders of Desk Side Support personnel. After contractors had built a computer, Desk Side Support

people picked up the PC from the inventory room and installed user-specific software that was not installed during the build stage. Then they brought the computer to the user and swapped the user's old computer for the new one. According to Edward, the operation manager, the initial plan was to have Desk Side Support people upgrade a few users each day. Nevertheless, it was quickly discovered that the Desk Side Support personnel, whose daily responsibility was to provide technical support to one or more floors of users, were having difficulty finding down time from their daily routine to perform upgrades.

The project was off to a slow start. To make matter worse, the gradual upgrade also led to a new issue. With some users already upgraded to Windows 2000 and Office 2000 and others not, some users were having trouble sharing files with their colleagues across platforms. All of which put more demand on the Desk Side Support people.

> *Initially, what I do is each one of my Desktop Support people is assigned a specific floor and they support that floor. Any issue that goes on, on that floor, they're responsible for. So, what I ask them to do is in addition to their normal support duties they were also going to rollout Windows 2000 to their floor. So, they were responsible for getting everyone upgraded. Well, it didn't work out so well. …. Probably the main reason was, they were too busy. It's kind of a catch-22 because we were still on a couple of different operating systems on the workstations on the floor.*
>
> – Edward, Manager of Operations Group

Another issue that took away Desk Side Support people's ability to concentrate on the upgrade was the application of service packs to multiple versions of operating systems that were

still in use. Without a centralized distribution capability, Desk Side Support people had to visit each desktop to apply the patches. For the operations manager, Edward, the solution was obvious, "We really needed somebody whose sole focus was on the rollout and allowed the normal support people to do their normal support duties."

He hired a young college graduate, Nellie, who had interned with CP, to concentrate on the Windows 2000 rollout. With the help of two other contractors, she completed the deployment shortly after. "I just went kind of full force into it. And I guess, I don't know, between 150 to 200 users, we'd done over a period of four months," said Nellie.

5.7.1.1 Desk Side Support Personal Strategy

During the Windows 2000 rollout, each Desk Side Support person had his or her own strategy on how best to bring the computer to the users. For instance, Nellie would first contact the user and set up a time to switch the computer.

> *I wouldn't just kind of show up and say, "OK I'm switching out your machine now." Because I wanted to fit it in to, I wanted to give them a chance, a time to get acquainted with it. Some people would just go and switch out the machine, and all of a sudden when the person comes back to their desk and all of a sudden and they've got this brand new machine and they don't know where it came from.*

Nellie understood the anxiety some of the users had when the computer that they were comfortable with was taken away. To her, being able to present and guide users through the first few steps of using the new computer eased the transition. She explained the mentality of some

users, “You know, they don’t want to give up their old machines anyway. They’re very comfortable with them and they don't want to change them.” Thus, she felt compelled to work every closely with the users and make the upgrade as easy as possible for them.

To prepare her users for the upgrade, Nellie sent them e-mail ahead of time to make sure that all software requirements were met and all applications were loaded before she brought the computer to them. The goal was to make sure users had everything they needed after the upgrade. Another Desk Side Support person, Susan, took a similar approach. To her, getting the desktop as close as possible to what the user had previously was the key to reducing the volume of complaints.

> *I think if you can get it just about like they had it you won’t get that many calls. But, if you go and just say “Hey, I’m just going to get you standard and we’ll work with you on getting the other stuff on and stuff like that, you’ll get a lot of calls. I didn’t get a lot of calls but you can, if you just put the standard stuff and don’t put everything that they have on there you will get a lot of calls.*
>
> – Susan, Desk Side Support

From the IT perspective, the majority of users were receptive to the upgrade. According to Nellie,

> *It is interesting; some of them absolutely love it. Because, you know, this is great, this is a new machine. And some of them were less enthusiastic. People were comfortable with what they currently have, something they get used to and so they are kind of less than*

thrilled about having to give up what they had because you know a lot of people have their set way of doing things. So, overall it was a good reaction.

However, Desk Side Support personnel agreed there were a few users who were reluctant to receive the upgrade and wished for it to be postponed. Susan explained some of the reasons behind the fear and the steps she took to calm the nervousness. “They're thinking, things won’t work the same. …They didn’t want to lose their data and wanted everything to continue to function as it had been. …Some people had had problems with certain programs and have finally got it working, and now all of a sudden you want to change so basically…you get a little nervous.”

For those users, Susan reassured them the upgrade would go well and worked with them closely to make them feel comfortable.

You have to ensure them that you can make this as comfortable as possible. Make sure all the applications that they were working with are working, where the data that they had was copied sufficient. Whatever they were working on, you have to just ensure that all of that is on the system and make it as comfortable as possible for them.

During the deployment, one issue that Nellie encountered was users’ failure to back up their data. Before the upgrade, Desk Side Support sent e-mail informing users to back up their data to the network share point. For users who were nervous about doing it themselves, Nellie volunteered to help them. At CP, there were two communities of users, laptop and desktop. Laptop users saved data on their hard drive. Because of that, Desk Side Support people were

more careful in making sure the data were backed up. For desktop users, there was a policy that they should store information on the network, which was backed up every night. However, some users still stored data in their hard drives and information got lost during the upgrade. “I have people who would come back five days later and say ‘Oh my gosh where’s my data?’” When that happened, Nellie explained she had to modify her procedure by providing an extra reminder.

> *I started to tell the desktop people to backup their data…if they had anything on their C drive. I didn't start doing that, but then … a couple of people who came back saying, where is my data? Well, you are not supposed to be saving it to your C drive anyway, you are supposed to save it to your network drive. So I finally said if you’ve got anything saved on your c drive please put it on your network drive. …That is your primary source of saving data.*

IT personnel learned from their experience and improved the process as the upgrade progressed. To cope with the potential loss of data, the planning manager made it a policy that all computers retrieved from the users would be kept for a week after the swap.

> *What we did is we saved the old PC’s for a week. So just in case we had to go back and retrieve some information that might not have been copied over, the PC was still intact. …But once their PC is re-pulled back for a week, then we wipe it clean and then reinstall the software..*
>
> -- Nellie, Desk Side Support

Fearing the data loss was probably users' main concern. Some nervous users requested to keep their computers, especially laptops, for awhile after receiving the upgrade. In most cases, Nellie would keep the computers for them and agree to keep it longer if they needed her to do that. Nevertheless, she admitted there were exceptions in which she allowed some of the trusted users to keep it for awhile. Her main concern was users would not migrate to the new computing environments when the option was given.

> *We have an instance where somebody kept her old machine and was working completely on this old machine and this new machine was sitting right next to it. And, if you're not going to use your new machine, you won't know what your new machine is. Does that make sense? And, I kind of forced my users to give me their old machine because I didn't want them using their old machines ...*

Nevertheless, not all users were wary about loss of data. In some cases, users were confident and had no problem giving up the computer. "And I even had some people who said, 'I know I've got everything, if I don't it's my own fault.' And they let me turn it in the same day," said Nellie.

During the rollout, users were given two laminated "cheat cards" that had information on the new e-mail system, Microsoft Outlook, and other Office 2000 applications. IT management felt Windows 2000 had a very similar interface and users did not need formal training. In the event where users were uncomfortable, the Desk Side Support people usually acted as trainers to get them up to speed. After the rollout, if users had any questions, they could call the help desk system and report problems.

5.7.2 Deployment at the Manufacturing Facilities

The deployment of Windows 2000 at the manufacturing facilities was a joint effort between the remote sites and the divisional headquarters. Before the upgrade to users at those facilities began, many back-end infrastructures needed to be established. The Windows 2000 upgrade at the manufacturing facilities was more difficult to coordinate because of the geographic distances. Prior to the upgrade, Luke came up with a checklist for the facilities. Approximately six months of preparation was required. "We had to have a lot of things in place before we did Windows 2000. We had AntiVirus solution in place. We had to have a name service in place, TNS. We had to have WAN connectivity because we just bought a company and we weren't even connected to them yet," Luke recalled.

One of the essential tasks that the manufacturing facilities needed to do was the installation of domain controller and creation of user policy. To ease the workload of local IT people, Luke and Nathan, a member of Projects and Design group at CP, built the domain controller and sent those to the facilities prior to corporate IT's arrival. All that sites needed to do was to find a place on the rack and mount the controller. However, some tasks remained the responsibility of the manufacturing facilities, like testing facilities specific applications.

To coordinate the upgrade, Luke flew IT personnel from remote locations to headquarters for meetings. He also visited various locations to meet with local IT personnel and discussed the upgrade with them. Throughout the upgrade, Luke held a biweekly conference to exchange ideas and lessons learned about deployment issues. The conference calls were helpful for IT personnel at different locations. Aside from that, a weekly report of completion status was published and given to all parties involved. The report was a great motivator for the manufacturing facilities to

complete the upgrade by showing the manufacturing facilities where they stood compared to others.

> *We would email this chart out every week and … it almost became a race. You didn't want to be way down here when everybody else was up there. …We had one [manufacturing facility] and we would send out the chart and everybody in the company would see that [manufacturing facility one] or whoever or [manufacturing facility two] was behind and so it kind of put some pressure on them without being rude or mean. … Just send it out every week and they would look at the chart and the manager would go "Oh, no I'm behind. We need to get our guys on this."*
>
> -- Luke, Project Leader

Besides having to manage and coordinate the project, Luke also personally took charge of the deployment at remote sites. The deployment at the manufacturing facilities was much more complex than the upgrades at divisional headquarters. For one thing, the sites were spread across different parts of the country. Many of these facilities manufactured different products, varied in size, and had different management styles. The manufacturing facilities were accustomed to managing their units without having much contact with corporate unless they needed assistance. The working relationship between corporate and manufacturing facilities was distant. One IT personnel, Nathan, explained, "In the past we never would even go and visit those manufacturing sites … unless they wanted us there."

At some sites, corporate personnel were perceived as "overhead," people who wore ties, handed down commands, and did not generate income like the people at the manufacturing

facilities. Luke explained that one of the goals that Sonny had for him was to change that perception.

> *When I came over here, [Sonny] sat me down in his office and he said "a lot of the [manufacturing facilities] dislike us very much because they view the main corporate headquarters as one that just tells them what to do." …He wanted to change the perception of our group to let them know we're on their side and we're not some big monsters trying to tell them what to do.*

During the planning, Luke was careful to not make the sites feel like corporate was telling them what to do. Luke explained that if a site didn't want to upgrade, he probably wouldn't force them to do that. But he would exert some pressure to see if they were willing to go along. Luke sold the project as something that could benefit the manufacturing facilities' backend operations in the long run.

> *Microsoft's selling point is that it doesn't really help the user that much but it helps the background administration. We can administer more machines now with less people. And that's pretty much the one driving factor of Windows 2000. I can click a mouse and install software on all 10,000 of those machines. And we used to have people run around and have to install, let's just say, a new Internet Explorer comes out and we need to roll it out and fix some problems. I can do it all from right here.*

Essentially, corporate wanted all sites to adopt Windows 2000, making the upgrade more of a "voluntary mandate." Nathan explained, "Well, you can almost say that we did impose that. The corporate pretty much said, Ok, this is the standard and you will be on this standard by this date. With that being said, we'll help you with it. You do it any way you want to do it but you have to get it done." As it turned out, none of the manufacturing facilities objected to the upgrade. However, depending on the facility, different perceptions were observed. At a site that was using Windows 95, the upgrade was welcomed with open arms.

> *It was completely receptive from the IT perspective…. I was tired of that Windows 95 architecture anyway and IT manager agreed with it. Definitely, at some [manufacturing facilities], they barked at anything that's pushed down to them at all initially until it's proven to them …Nobody at any of these [manufacturing facilities] probably is going to say "Ok, you said to do it, then let's do it." Most of the guys are at least going to ask questions and say "Why, what's your justification, are you sure it's going to work reliably, etc."*
>
> – Kenny, Network Administrator at Manufacturing Facility A

While there was consensus that there were many advantages to migrating to Windows 2000, there was some resentment of the increase of workload placed upon the facilities.

> *I really think the upgrade will be beneficial in the long run. I understand from the corporate mindset that they believe that if everybody walks and talks the same way that life will be better, maybe. And, it's probably easier in the long run to roll out software to*

individual PCs that everybody's on the same page. But what really makes life easier for the folks in [headquarters location] doesn't necessarily mean easier for the people in [manufacturing facility B]…Our team here still has to do everyday mundane work of supporting the [manufacturing facility] environment, in addition to the initiatives (headquarters location) says that we have to do, like everybody's set up a newer release operating system or a newer computer. We have to do our own job in addition to trying to keep the [headquarters location] crowd happy.

- Hugh, IT person at Manufacturing Facility B

Unlike IT personnel at division headquarters, where there was a clear division of labor, IT personnel at manufacturing facilities were "jacks of all trades" in many cases. Some of them were full-time IT employees; others were accounting managers who also took on the systems management tasks. While they typically made decision for plant specific software, they relied on the expertise of IT personnel at division headquarters to conduct the research and keep them up to date on the latest technology. Even though the upgrade added work to manufacturing facilities, one site explained, they still complied because they were part of the company.

Again we're all part of a big team in a way but it's sort of a give and take thing. We can allow them to do things and sometimes they do things we don't want them to do but when it really gets down to it, we still have to reluctantly agree.

- Hugh, IT person at Manufacturing Facility B

Corporate IT people understood the unwillingness at some of these manufacturing facilities. According to Nathan, one of the Projects and Design personnel who had been to the manufacturing facilities, the upgrade was an added task that was put on the manufacturing facilities. “Well, everybody these days they already have enough work to do so when you bring them something else to do there’s always that resistance.”

To assist manufacturing facilities, corporate IT provided manpower and technical expertise needed to deploy the upgrade. Luke assembled a small team of IT contractors and personnel to go with him to remote sites. The number of people Luke took with him depended on the number of IT personnel that manufacturing facilities were willing to spare for the upgrade. “I matched them one for one,” said Luke. The approach was meant to encourage participation from the remote sites. Furthermore, being aware of some criticisms that field people had of corporate personnel, Luke brought only the most qualified IT people from the divisional headquarters with him.

> *So it was a big risk, us going out there. And that’s why I got the best people I could find. …We have Desktop Support, we have Servers Support, and we have Project Team, which is what I’m on. Most of the time I took project level people with me. And that is our most technical people. And this isn’t rocket science work rolling out PCs, but I wanted them to get the very best impression of us here.*
>
> – Luke, Project Leader

One impression that division headquarters wanted to avoid was being seen as “a small army” who came over to take over the project. The position that headquarters took was to allow manufacturing facilities the freedom to organize the upgrade anyway they wanted and to deploy

in a time frame that was comfortable for them. The only provision was that the upgrade had to be completed by the deadline.

> *Our position with this Windows 2000 rollout was we'll help them anyway if they want us to help them. We're not going to force them to do anything. We're not going to force them to do anything a particular way. They can do it any way they want to as long as they get it done but we would help them if they wanted help at any point in time.*
>
> – Nathan, Projects and Design Group

> *We're just there to give them a hand, as an added resource that was dedicated just for the rollout. And some of the site are a little apprehensive that the 'corporate' is coming out to help them. And they were under the impression that we were just coming out to tell them how to do it. But, the policy that we had was we were here to help you, it's your [manufacturing facility], you tell us what you want to do.*
>
> – James, a Contractor

During the upgrade, Luke emphasized the need for remote sites to participate. One reason was corporate IT personnel were not familiar with the situation in the plant and couldn't deploy the upgrade efficiently without the help of local IT people.

Geographically, most of the manufacturing facilities were located "in the middle of nowhere," as "islands unto themselves." The facilities were generally large in size and occupied many acres of land. To get from place to place, corporate IT people would ride around on golf

carts and bicycles. The working environments at the manufacturing facilities were quite different from the headquarters where most employees dressed business casual and sat in offices.

> *We wore t-shirts, blue jeans, hard hats, and earplugs. You have to wear this earplug. There are chemicals in these [manufacturing facilities] and you have to carry this mask. If there's a chemical spill you have to put this on…We had to go through safety training to go to these [manufacturing facilities.] We don't have to worry about a forklift running into us here in [headquarter location], but when you're walking around the [manufacturing facilities] you got forklifts going around…*
>
> -- Luke, Project Leader

In general, corporate IT people stayed at the remote sites from three days to two weeks. Depending on the manufacturing facilities and how the local IT management wanted the upgrade to be done, slightly different approaches were used at each site. Many sites adopted some variation of the "build and swap" approach. Nathan recalled the approach used, "Sometimes they put us in one room and, all we do is just build machines. Sometimes we actually physically go out and roll some of the machines out." If the facilities did not have enough computers for IT people to build, they would go to users' desks and install the operating systems on the spot.

IT people could build approximately four to eight machines at one time with enough network ports and monitors to attach to the CPU. In one week, they could build around 200 to 300 machines, but corporate IT explained there were times where they fell short of this. Low volume deployments were attributed to lack of preparation, unusual practices, or unforeseen circumstances at the site that increased the length of upgrade. One remote site stood out in James' memory.

I hate to say but …both [Luke] and I felt that they were very disorganized up there and they have a union, which was kind of interesting. The only people that could actually touch the computers, pick the computers up were the union representatives. So, the week that I spent up there was very … I don't want to use the word wasted time, but it wasn't very productive because we only,…only cranked out only 3 or 4 machines 'cause they … were trying to personalize them as we were building them.

Another IT person who went on the same trip echoed that, "They weren't really prepared and they were kind of dragging their feet on the upgrade." The project leader explained, the site expected corporate IT to take over the project and didn't provide any assistance at the beginning. In the end, he had to provide guidance on how he thought the process should go. Corporate IT people didn't accomplish much during their first trip and returned at a later time to finish the rollout. However, corporate IT people also had some good experience with remote rollout; some sites were well prepared and others actually got the work done prior to their arrival.

One difference that James noticed between headquarters and remote sites was they did more customization for users at the remote sites. Unlike in headquarters, in which Desk Side Support responsible for their users, here IT people from headquarters would play the role of both build and swap. Of course, it really depended on how the site wanted to do it. Some wanted to eliminate the impact on users. For example, an IT manager at one remote site requested IT people from headquarters to put the user's family photo back onto the computer.

When I said extensive customization, things like desktop images, little personal touches on their PC. The IR staff at the [manufacturing facility] decided to ease the impact of end user, we will set up the desktop image of their kids back on the new PCs, or copy their favorites over, double check their favorites make sure they all work. Just a number of things like and we were, we were doing that to ease the shock of the end user.

– James, Contractor

The goal was to make the new computer look as close as possible to the old one. By closing the gap of difference between the two computers, IT personnel hoped to minimize the impacts that users felt from the changes. In the next section, the impacts that users felt from the upgrade will be discussed.

5.8 Impacts

5.8.1 Impacts of Windows 2000 Upgrade on Users

The Windows 2000 upgrade project was more than just an operating systems upgrade. During the upgrade, IT also migrated users from Office 97 to Office 2000, and switched users' e-mail application from MS Exchange to MS Outlook. Thus, the impacts that users received from the upgrade were not limited to operating systems. Two categories of impacts were identified: impacts from the upgrade process, and impacts from changes in new software like Windows 2000, Office 2000, and the new mail application.

5.8.1.1 Impacts from Upgrade Process

During Windows 2000 deployment, IT was careful to plan for a smooth execution of the upgrade. The “build and swap” approach was used to reduce the time spent on conducting upgrades at users’ desk. IT people also strived to make the transition process as seamless as possible for the users by attending to their needs in advance, for example, making sure all additional software was installed. Despite the best effort, some users experienced lost files after the upgrade. According to one user, Danny, “They gave me notice and they said that anything that I wanted to keep I needed to put out on a, like a personal drive, which I know where it’s at. So, I put stuff out there and somehow it got lost.” Danny was lucky because the IT people did find his file after a few days. Nevertheless, it still created some inconvenience for him. “It was kind of like you work on all this stuff and you wanted to keep it, and it's like it's gone. It's a big deal. …but yeah I definitely got it back.”

In another case, Isaac, a user at a manufacturing facility recalled his inability to find some of the folders immediately after the upgrade. In his case, some of the files were still missing when I interviewed him.

> *After we made the first upgrade …I did have a problem finding some of my folders and my files. …I would have to kind of look around in the system and try to find them cause the upgrade placed my files in different folders and subfolders under different primary folders. …In fact there are a few files that I have yet to find. I don’t know if they were just lost in the upgrade or if they’re just buried so deep I haven’t been able to find them yet.*

For majority of the users, the overall upgrade went smoothly. “As a user, quite honestly, they did a great job,” commented one power user. Other than lost files, another issue that some users experienced was user’s inability to share files. Because not all users received the upgrade at the same time, there was a period of time in which some users had migrated and others had not. Even though users of MS Word and MS Excel were able to open files in both versions, users of MS Access were having trouble sharing databases.

5.8.1.2 Impacts from Software Changes

The impacts that users perceived from the software upgrade were usually dependent on the degree of change in the software functionality that they used. Users who used standard features experienced less impact because most basic functionality remained the same even if the vendor added changes into the new version.

In general, most users did not recall having huge impacts from the upgrade. “I think to me …they made the upgrade and for me it’s business as usual,” said a user from one manufacturing facility. Another user, Ben, recalled some minor modifications in the interface, “It’s a little bit different as far as the interface went but it was still easy enough to understand. It wasn’t a major issue at all.”

Overall, business users who used mainly Word and Excel did not perceive much change, but users who worked with Access database had a different experience. The most obvious change to users at all levels came from the switch of e-mail application. The next few sections will touch on the impacts of software changes as seen from users’ perspectives. First, the changes in operating systems will be discussed.

5.8.1.3 Impacts from OS Upgrade

From IT perspective, users who were on Windows NT 4.0 should have found Windows 2000 familiar. Hence, the operating system upgrade should have little impact on users. “It looks the same for the user, “Start,” “Run,” it’s like Windows 95/98, NT4.0 … they can care less,” said the project leader. An IT contractor echoed that, “…So it's the same log-in process. It's just that they have new user name and new domain to login to.”

Overall, changes from the operating systems upgrade were fairly minimal to the user. The largest noticeable change was the way they logged in every morning. “The login security feature seemed to be a little different. The login is different here than it used to be,” commented one business user, Wilson.

Other than the log-in changes, a few users also noticed that computers seem to perform slower at certain functions. According to an IT user who manages logistic systems at a different department, he noticed the performance of Windows Explorer has gone down. “Every time I go click on to save a file I can literally go do something else and come back before it drops down my box because it’s trying to understand where all my drives are mapped to and trying to synchronize files.” His situation was fairly unique because as a system administrator, he had many network drives that he was connected to. At the same time, he acknowledged that offline file synchronization, a new feature in Windows 2000 that synchronized users’ files on the hard drive with their files on the network, was a beneficial feature.

Nellie, who was responsible for rolling out Windows 2000 to users at headquarters, recalled that some users were skeptical about the receiving new operating systems after hearing rumors about its performance. “For the most part, they were like ‘I’ve heard stories about this new operating system’ and from a technical standpoint, the biggest complaint about the operating

system is that it's slower to shut down," said Nellie. The slower shutdown was again attributed to offline file synchronization that goes into action the moment users click on the "logout" or "turn off computer" function. Nevertheless, one user, Calvin, commented on the quicker boot up speed that he noticed switching from Windows NT 4.0 to Windows 2000. "What most people perceive is it boots up faster and they like that. It took NT awhile to boot up once you had turned your machine off and turned it back on," Calvin explained. For some of the users at the manufacturing facilities that were still using Windows 95, after they received the upgrade, one of the visible benefits was their operating systems no longer crashed once a day. **"**One of the biggest things was stability. It was reliable, it did not crash and using the old software compared to Windows 2000 was like night and day. There's a big difference," said Lance at manufacturing facility A.

At some of the manufacturing facilities, the system administrator took the advantage of the Windows 2000 security features and restricted users from performing system activities. After the upgrade, many users found themselves stripped of their capability to install programs and in extreme cases, unable to change the time clock on the computer. "It's very inconvenient," said Ben, "I could understand to a certain extent why it's done that way. You may have some people that are not computer literate enough that it could cause you problems but for the most part the power users, what does it hurt me to set my clock?"

The change certainly affected power users more than standard users because power users were the one who used to have more privileges in the past.

> *"I don't necessarily want to have free access to delete files and install other programs, but what Windows 2000 does with the new file security prevents an installation of any software. So I could not install anything without the approval and support of IT*

…Obviously I don't like it but I do understand the reason it's there. I obviously respect it."

– Lance, Business Manager at Manufacturing Facility A

To Lance, the inability to install software meant he had to rely heavily on IT people to get the needed installation, which was less convenient and more time consuming.

"That's the biggest problem, then, I could fix it myself at whatever timeframe I had but now I have to rely on them. So, I'm trying to push them to help us quicker. I mean just having to call somebody takes more time than me being able to check a setting or let's say if I could be able to do something I could go in and change that and then we're fine if it worked."

Even though users explained they understood IT's reason for not allowing them to install software, they did not like the fact that their freedom to install screen savers or change the time on the clock was taken from them.

5.8.1.4 Impacts on Office Applications

To most users, application upgrades to Word, Excel, and PowerPoint did not bring much impact. "All your other Microsoft applications, Excel and Word were functioning all the same. If there were new features, I haven't seen them yet. I just did the basic things in Excel." said a user. "Clip art in Power Point would have been different, I think," said another user.

Many users saw some changes in the interface but explained that they were able to navigate as usual. One user, Danny, expressed that he was pleased with the changes, “Yeah, we used Microsoft Office 97. So, we changed it to 2000 which is nice.” Overall, while users recalled some change in “look and feel,” the consensus was that the modifications could be tackled without much difficulty, at least in Excel and Word.

The upgrade to Access 2000 presented a slightly different scenario. Access 2000 was quite different from Access 97. One of the changes is in the user interface. According to a user, Peter, “The appearance changed. If I went into Access, whatever the old version was versus 2000, you had them up on two screens you could tell a big difference.”

Another problem for Access users was the lack of file compatibility between the two versions. In order for users to work on databases created in the previous version, a conversion to new version was required. Otherwise, the previous database remained a read-only file to the application. One user, Isaac, observed, “I know we had some folks in the [manufacturing facility B] that had some issues with Access files that they had produced in the past, and they had spent a lot of time making those work in the new upgrade.”

To complicate the matter, not all users received Windows 2000 upgrade at the same time, which created a file sharing problems among Access users. One user, Ben, recalled the difficulty he encountered and the tactic he used to cope with the situation.

> *If you’re in 2000 and you try to work backwards it won’t work. So, you’ve got to upgrade that old file. You’ve got to modify it to the new version. That was a problem. A lot of the stuff we use is shared. We have a public server, public drive and a lot of the files were shared and for me to work on a shared file I would have to go get it, save it on my drive*

as a different name, modify it to the new version and do my work then, and then send it back down and save it back in the old version, old format. That was very inconvenient.

As a temporary solution, IT installed both versions of Access for users so they could continue to work with files from Access 97 and Access 2000.

5.8.1.5 Impacts on Switching of E-mail Application

To many users, the most obvious impact of the Windows 2000 upgrade was not the operating systems change or the Office applications upgrade, but the switch of e-mail applications from MS Exchange to MS Outlook. Even though both applications were Microsoft products, the switch was not equivalent to same-product upgrade. Many users noticed the differences between the two applications almost instantly. According to a user, Ursula, who also happened to be an IT employee, "The main thing is the e-mail. Everyone uses e-mail so I was like …'Where's my calendar now?' or 'How do I go and send a new message? How do I setup my address book now?' and that kind of stuff… It's quite a bit different."

Many users favored the new e-mail application citing various advantages. "I could pre-send my e-mail if I know I'm going to be in late tomorrow morning and it's something that I need to send to somebody in the morning I can go in, type up my email and say do not send this before 8:00 in the morning," said a customer service representative, Jane.

Another user, Isaac, from the manufacturing facilities explained that he really like the changes, "Another thing we use often is Microsoft e-mail and I like the new version that we're using better than I did the older version. It took me a little while to figure out how to use it. Some of the new features on it but I like it a whole lot better than what we had before."

One feature that users especially like was the ability to schedule appointments and look at other colleagues' calendar. "I like the meeting planning schedule piece of it that we can plan the meetings with the other employees. That's real handy. We can go in and set up a meeting with four or five employees and actually view their calendar as well, and then pick the best time to hold our meeting," said a user.

However, not all users liked the new e-mail application initially. Peter preferred the previous e-mail application but explained he eventually became accustomed to the new application.

> *I think it was Exchange then went to Outlook. I liked Exchange I guess cause the prior jobs and companies I worked at have always been on Exchange. Then going to Outlook it's just different and I guess you're used to using one application versus the other and I preferred Exchange. I think it's a little more straight- forward and easier to use. But Outlook it's nothing difficult, it's getting used to the way how it is and how that works.*

Users of e-mail also encountered the issue of not being able to access each other's calendar due to different versions of e-mail during the deployment. According to Kellie, an IT user, "One issue we did have actually we all did not get upgraded at the same time. Some of us were still on 98 and others were on 2000 and we have access to see each others' calendars, check each others' schedules, and book meetings. Well, if we weren't on the same operating system we lost that feature. So if I was still 98 and I was trying to find somebody's calendar in 2000, I couldn't do it." Kellie also recalled losing an appointment as a result of logging in from different versions. "I went to a training class, I logged on to another computer and here I was on 98 and

there I was logged in to 2000. And I tried to get to my Exchange account, your calendar would get deleted actually. And, you'd come back to your desk and all of your appointments were gone." She continued, "But I remember that being a problem for me 'cause that happened to me where a lot of my appointments got deleted. …I mean you could log on but just as long as you didn't go into the calendar, you were ok. …But, that's the only problem that I recall from changing the operating system."

5.8.1.6 Learning Curve

One common impact that almost all users experienced during the initial stage of the Windows upgrade was the learning curve. According to one power user, Ursula, "The learning curve is getting use to the new features. To learn new feels of the software so you have that learning curve that takes awhile to get through." However, many users were able to adapt to the new technology after the transition period. One power user, Tom, explained, even though the change he experienced was minor, there was still an adjustment period. "Yet, I had my minor little complaints but that's just because of the change. Everything doesn't work just like it used to. That's not necessarily a bad thing. It's just getting use to things. But overall, the change was very positive."

For this particular upgrade, users did not receive any official training. IT management felt the changes were manageable, even though they acknowledged that the e-mail applications deviated quite a bit. Just two laminated cheat cards were distributed to the users when they received the upgrade.

…they went from the NT platform to Windows 2000. Basically it looks alike but there are some differences. They went from Exchange mail system to Outlook, which was considerably different. So we made sure that each person that got a new PC received some cheat sheets to help them. We thought that might cut down some of the help desk calls.

– Zack, IT manager

One user, Mary, recalled receiving that, but explained she hasn't had time to read the cards. Another user, Danny, explained he only use the cheat card to look up information on the new e-mail applications. "Play around with it. That's the easiest way to do it. …Generally, playing around with it will do it." Overall, users preferred to just dive in and learn by working with the software.

And, my thing is I'm the type of person, hands-on is better for me. If I don't smell smoke from the computer I pretty much figure I'm not doing that much damage so I kind of like hands on. Cause it's easy to sit someone in a class and say this is what you do but hands-on is a lot better for me cause when I get there I can actually understand what I'm doing

- Jane, User

I'm not scared to push a button. I'll try, I'll play because I know they're not going to take functions out of a program. What I'll do is I'll play with it till I figure out, I'll use the help screen, I hate to admit that, but I'll actually use help to figure out what I want to do. But I'm not scared to try things.

– Ben, user

Usually, if a user could not figure out how to perform something by trial and error, they sought help through on-line documentation. Another user, Isaac, proclaimed, “I’m someone that’s very familiar with the help files.”

Even though help desk service was available, many users explained they seldom called the service unless it was necessary, “I’m going to try to solve problems myself before I call somebody else,” said a user who did EDI. He explained that he had been working with computers since seventh grade. Another user, Ben, preferred seeking assistance from their colleague and felt that the help desk was for beginners. “I’ll work fairly closely with the programmers …and they got one lady over there that’s she’s super good at Access. If I had a problem, I called her. The help desk is for the, I don’t want to offend anybody, the average user or the beginning user to call and say ‘Well, I can’t seem to do this. What do I do?’”

For many users, Windows 2000 was just another upgrade of Windows operating systems, and they did not feel much impact from the upgrade other than some interface changes and the learning curve. For IT personnel, Windows 2000 upgrade brought tremendous impacts to the way they worked. The impact of new operating systems was felt from Desk Side Support people who dealt with users at the front end, to the system administrator who managed the server at the backend. Many of these impacts were a direct result of the technology improvements in the new operating system. Other impacts were related to the time, effort, and chaos involved when conducting an upgrade. Still, others were the result of the decision to bring all users onto Windows 2000. Each of these impacts is discussed in greater detail in the next paragraphs.

5.8.2 Impacts of Windows 2000 Upgrade on IT

Windows 2000 is an extension of Windows NT 4.0. Originally termed NT 5.0, the operating system was renamed Windows 2000. While Windows NT was a more reliable operating system, it lacked the “plug and play” feature available in Windows 98. This feature was especially important to laptop users who had to travel, use docking stations, and connect to different printers. In Windows 2000, Microsoft integrated the traits of Windows NT and Windows 98, into one operating system. For the first time, the stability and security features in Windows NT were available along with the user friendliness of Windows 98.

Although Windows 2000 came packed with many features, it also put a lot of demand on hardware requirement. Even though many of the users at CP had hardware that met the minimum requirements of Windows 2000, some users from acquired companies were still on older machines. In order to implement the upgrade, IT people had to put in more powerful computers. At some of the remote locations that were still running on Windows 95, more than half of the hardware was replaced. Throughout the division, every user received a hardware upgrade along with the operating systems upgrade. Depending on the job requirements, some users received brand new computers, others received upgraded machines.

In addition to the cost of hardware acquisition, additional manpower was hired to help configure PCs during the peak of implementation. Replacing and refreshing hardware were not the only preparation needed. Like all IT projects, a successful implementation of Windows 2000 took a lot of planning and preparation before the upgrade of individual users can begin. IT people also had to put in back end infrastructure. In addition, all division-specific applications were also tested to ensure compatibility with the new operating systems. Although none of these costs were calculated, they amounted to a substantial figure.

5.8.2.1 Improved Operating Systems

Immediately after the Windows 2000 upgrade, IT people noticed the new operating systems was a more stable platform compared to its predecessor. IT people found Windows 2000 "crashes less" and "works better with our hardware." According to the planning manager, Zack, "That right off the top of my head is one of the largest advantages because we were always replacing PCs." Prior to the Windows 2000 upgrade, one of the problems that IT people faced constantly was the instability of previous operating systems. "We would constantly have system crashes," said Zack. "Something, either software wise or hardware the NT platform didn't coexist with, and it would automatically shut down your PC… we would get what they call the blue screen of death. Well, the screen goes blue. And it's like nothing works."

System crashes were not only costly, because IT people had to replace users' PCs, but also time consuming. Having to swap the PC is just a part of the effort to restore users' computing capability. The manager explained that a tremendous amount of time was spent on recovering lost data and putting it on the new machine. "The technical people would be spending hours because when it crashes you had to try to restore all the files, get the data and it was literally a nightmare …" With relief, the planning manager explained that the situation improved after the Windows 2000 upgrade, "We would always have to rebuild a PC, give them something better. That has disappeared." An IT person, Susan, who supported users, provided a similar account. "It crashes less …It has less problems…after the upgrade … it just appears to me that it has smoothed things out a little bit."

In addition to being a more stable operating system, Windows 2000 also had "self healing" capability. In the event an illegal installation or deletion of an application on a user's desktop, the operating system automatically detected the problem and repaired the application.

IT management was especially delighted with the self-healing application. The new feature not only reduced the chance of system failure when a user committed an illegal operation, but it also reduced the need for IT support in response to those problems. Even though the new operating systems did not eliminate all technical issues, technical support was made easy through a tool called Net Meeting.

5.8.2.2 Support Features – Net Meeting

"The other benefit that we saw from windows 2000 was the enhanced tool set for support group," said the senior manager, Sonny. The tool that the senior manager referred to was Net Meeting, a technology designed to facilitate collaboration between multiple parties and was often used by ordinary users to teleconference and share applications. Net Meeting was not a brand new technology. It was available as a separate component in the earlier version of operating systems but came standard with Windows 2000. "It's actually part of the operating system," said Luke. "It allows you to remote control somebody's PC."

IT personnel took advantage of its desktop sharing capability, and used Net Meeting to troubleshoot users' problems. One IT person explained, "If somebody in a different building calls, we can just say 'Take your hands off the key board. We're going to remote control your machine to change a setting or fix something.'…So, that saves us some leg work." Many IT personnel welcomed the convenience of having access to a user's desktop without leaving their workspace. "I use that and I absolutely love it," said one Desktop Support person, Nellie.

To IT management, the best part of using Net Meeting was that it allowed the support personnel to identify a user's problem with little ambiguity. "What we discovered … is that

approximately two-thirds if not more of the time spent on help desk call is just trying to figure out what that person is talking about," said the senior manager.

By allowing desktop support personnel to see the problem as it was being described to them, IT support got to the root of problem quicker. "So it's saving the user's time and it's saving our support group's time," the senior managers continued.

5.8.2.3 Active Directory

The largest impact of the upgrade came from the changes brought by Active Directory, a new technology component of Windows 2000 architecture. While it is possible to adopt Windows 2000 without implementing Active Directory, many of the benefits anticipated in Windows 2000 would not be realized without this back-end component.

Active Directory was a leap in network technology integrating applications, users, and data into a centralized location. Serving as the "main switchboard of a network operating system," Active Directory allowed system administrators to manage computing in distributed environments. Although the benefit of active directory was not obvious to standard users, IT personnel found the new back-end architecture to be very impressive. One task in particular was the capability to distribute software over the network.

At a large organization like CP, applying security patches or application upgrades to users was labor intensive and time consuming. Prior to the Windows 2000 project, the standard way of applying software patches or upgrades to users' systems was by making personal visits to each desktop with a disk in hand and performing the upgrade manually. Thus, the new capability for IT personnel to automate the deployment of patches and upgrades was seen as, "one of the largest advantages," according to the planning manager. "Some applications like SAP "would

come out with an upgrade once a quarter, where we would have to go out once a quarter and hit every machine," he said. "We don't have to do that anymore. And, that would always be like a week-long process," recalled another operations manager.

Even though not all software had a client upgrade every quarter, many other software applications like Norton AntiVirus, MS Explorer also required frequent updates of files. Each round of software distribution could take from a few days to several weeks to complete. The senior manager of the Infrastructure Support Group explained the time was dramatically decreased after Active Directory was installed.

> *Over the last few months, there's been a whole bunch of security patches, for OS, browsers, and Outlook…Through Windows 2000 Active Directory, we have been able to apply these patches automatically within, actually within half a day, we have applied the patches to 9200 desktops vs. weeks doing by hand… So it's been a huge impact there.*
>
> - Sonny, Senior Manager of ISD

The impact of Active Directory extended beyond the time saved on software deployment. Before Active Directory was put in place, software installation was conducted after regular office hours to minimize the interference of upgrade on users' routine. Zack could not help displaying his joy as he recalled how that had changed, "So I can tell you myself I have not been here on weekends doing that sort of thing anymore, ha ha ha (laugh)." Even though no official report showed how much time CP saved, the amount of overtime that IT put in was dramatically decreased, "I know that for a fact," said the manager.

Besides software distribution, Active Directory allowed system administrators to have a tighter control over users' desktops. Through the use of group policy and other technical

features, a system administrator could exert control from the type of application that could run on the desktop to what a user can perform on the system. “There is a lot more control over user policy and stuff, keeping them from loading unauthorized software; a lot easier in 2000,” said the IT contractor, James. After the upgrade, many users found themselves stripped of all privilege to install any software. While some of the power users were not thrilled with the changes, system administrators have found the undertaking necessary to enable better security for the organization.

What Active Directory brought to the table was not just the capability to remotely distribute software or tighten control on user’s desktop. The new technology created an architecture that integrated all network components together and revolutionized back end operation management.

Nonetheless, Windows 2000 implementation was not an overnight success. Besides the long technology governance process used to evaluate Windows 2000, all IT personnel at CP, depending on their job responsibility, attended between three days to one week of training. Like any IT project, Windows 2000 was carefully planned to make sure the deployment would run smoothly. Despite the best effort, the biggest challenge of Windows 2000 upgrade came a few months into the rollout process when Active Directory crashed. “We were about a third of the way rolled out, every single one of those sites that were rolled out in windows 2000, they were down. The whole [manufacturing facility] was down. They weren’t making [product]. It was bad, for about a week.” The project leader recalled that he and his team were in the field deploying Windows 2000 when the disaster happened. “The rollout completely stopped for about a two, three-week period. We had to cancel trips, we had to come home from where we were at,

we had to pack up and come home. It got very ugly. We had some ugly conference calls. People saying 'Somebody needs to get fired, we're losing millions of dollars a day.'"

As it turned out, someone in the field location had placed a large file into a folder that automatically "gets replicated all over the entire enterprise, from Seattle, Washington to Jacksonville, Florida and everywhere in between…ha ha ha," one IT manager recalled, laughing while telling about the incident. Of course, it wasn't funny when IT personnel were in the midst of it. "We had to do a lot of disaster recovery; we had to make a lot of phone calls, a lot of apologies. We had to get Microsoft in here to help us get it back up and functioning. It was just a mess for about three weeks," said the project leader.

The Windows 2000 project was a partnership between NTPS and CP. While CP was responsible for bringing the upgrade to thousands of users across the nation, NTPS was in charge of the back end infrastructure. Even though the operations of Active Directory would eventually be passed to the corporate operations group when the environment was considered stable by NTPS, corporate IT essentially owned the technology at the initial stage.

> *We are accountable for that platform if you will. So if anything happens my group's accountable for it during rollout… if they roll it out to the [manufacturing facilities] and there is a design issue or there's a problem with the operating system that impacts their business they're going to come to my team for an answer of why they're having problems."*
>
> - Joseph, NTPS team lead

Although the division and corporate IT were polite to not point fingers, slightly different perspectives were given when they talked about the incident. The project leader at CP felt corporate IT was still treating the project like a test and didn't give it the serious attention it deserved. "What it did to those guys is they realized they needed to get in gear and get this into production where they're monitoring it, where they know when something goes wrong before it happens instead of somebody calling from across the country saying "I can't log on, why not?"

From corporate standpoint, one of the problems was having too many IT personnel who had "admin. capability to make changes in Active Directory." Someone who had insufficient knowledge could easily make an error and create an outage. While acknowledging it was a learning curve, the project manager at corporate IT explained they had taken precautionary steps to prevent the incident from recurring.

First, they began to perform "frequent backups" on multiple domain controllers including the main database, several times a day. Second, they "greatly limited the people who have 'that' level of access within 'that' area of the active directory." Restricting the access to Active Directory to four people within each division allowed them to have better control over Active Directory. Third, they changed the approach used to manage the infrastructure, "Another action we're taking is to centrally manage the domain controllers within one group, instead of having that across all divisions." It took some adjustment to implement this new policy because CP put domain controller and applications on the same server, and its IT personnel would need access to the server. To solve the problem, corporate IT provided the division with a dedicated server to run domain controller so personnel from corporate IT could have full control over it.

CP ended up rebuilding all domain controllers for the field locations that were affected. Despite the mess that the incident caused, the project lead was glad that it happened early in the

upgrade process, because it could have shut down the entire division if it had occurred after all manufacturing sites had migrated to the new systems. It was a learning experience for all parties. After the dust settled, project lead reassured offices at remote locations to continue with the upgrade.

> *Finally, then I had to get everybody's nerve back up; okay, who wants to do the next rollout? …because people were, like, I don't want it now, I don't want it. So we almost had to resell it again saying "look everything's fine now, it's not going to happen again" and we haven't had anything happen since then. And they're doing a great job of monitoring the systems. There have been problems but they caught them real early and before it affects anybody.*
>
> - Luke, Project Leader

5.8.2.4 Impacts on Peripherals

Operating systems upgrades often bring changes beyond the operating systems. As a central controller of input and output, the impact of operating system upgrades often extends to its external peripherals. During the rollout, IT management upgraded printer drivers to be Windows 2000 compatible and users had to be re-educated on new printer queue names. Even though printer driver upgrades were not technically challenging to perform, they were nevertheless hectic events that many IT Support personnel recalled living through. "For a minute, the printer change was more high maintenance than anything," said Susan.

The change came in the midst of migration. Some users were still using Windows NT 4.0 and had trouble printing to the upgraded print server. "We had to go and download drivers on

their system to make them compatible, so they could print to this printer" said one IT support person.

Some users added to the chaos by taking the matter into their own hands, helping other users to connect to the new printers. They "probably didn't do it right," according to one IT support person. There was also confusion when a user sent out an e-mail message with misleading information that the print server for NT 4.0 was no longer available. Soon, help desk personnel were flooded with calls about printer problems. "At that point, the cat's out of the bag, so you just go ahead and you wait for the problems to come in and you go fix them," said the IT support person.

5.8.2.5 Standardization of OS

After the Windows 2000 upgrade, one of the biggest benefits that IT personnel reaped was having a unified operating system across the entire division. Having all users on one platform was a luxury that IT management had not experienced since the division went to Windows NT 3.5.1 many versions before. Before the upgrade was carried out, multiple versions of operating systems were used throughout the division.

> *You would go to somebody's desk they might have 3.5.1, one of them might have 4.0, they might have 95, they might have 98. It really depends on what floor they're on what group they're in, what [manufacturing facilities] they were at. It was all different. It was really a mess.*
>
> – Luke, Project Leader

One reason for having multiple versions had to do with the way upgrade was conducted at CP. Traditionally, operating systems upgrades were conducted on an as-needed basis. For example, an upgrade to NT 4.0 was conducted when a PC was replaced. Windows NT 4.0 migration was never carried out as a mass upgrade. Although Windows NT 4.0 had been the standard for a few years, many field offices including some users at divisional headquarters were still on Windows 95.

Windows NT 4.0 was also not installed on laptops due to its lack of plug and play capability. For mobile users who traveled and needed connection to external peripherals like printers and docking stations outside their home offices, Windows NT 4.0 was insufficient. Instead, Windows 98, a considerably less stable operating system was installed on all laptops to accommodate the special need. The landscape of the operating systems before Windows 2000 upgrade was a mix of Windows 95 and Windows NT 4.0 on desktops and Windows 98 on laptops.

With users on multiple versions of operating systems, IT personnel had to be knowledgeable on all versions of operating systems even when they were not the designated support persons for the operating systems. One IT administrator, Tom, who managed a logistic application, recalled having volumes of documentation just so he could support his users who were on multiple platforms.

> *I could give you examples of how much documentation I had to have; at least 80-something pages were the documentation for every operating system that we supported. So, from installing each application on Windows 95 to 98 documentation, NT documentation, and then troubleshooting, trying to make sure we had documentation to*

trouble shoot each one of the operating systems so it was definitely a big support nightmare.

IT personnel often had to first identify the version of operating system that was installed on the user's computer before they could proceed to trouble shoot. According to Luke, "Let's say a user calls up, 'I'm having a problem printing' or whatever. My first question: what are you running, Windows 95, Windows 98, Windows NT? If you're running Windows NT, what service pack are you on? Now that question is completely eliminated. We know exactly what's on their machine."

The standardized environment not only eliminated the need to support multiple versions of OS, it also cut down the time needed on software testing. An application went through rigorous testing before it was deployed. According to Tom, "It is tested on each operating system and each combination of applications the user might have." With one platform, there was no longer a need to test software on multiple operating systems. "When we test applications now, we don't have to test them for 95, for 98, for NT...we just have to test one operating system," concurred Luke.

5.8.2.6 An Unexpected Impact - A Problem and a Blessing

The Windows 2000 project also brought unexpected improvements in other application areas. The impact was not a direct effect of Windows 2000 technology, but it was a solution that allowed Windows 2000 project to move forward. During the Windows 2000 testing period, an application used by the logistic department was found to be incompatible with the proposed operating systems. In desperate need of a solution, the system administrator took up a casual

suggestion from a contractor who heard of the problem, and looked into a technology called Cytrix. The technology was later implemented, paving the way for the Windows 2000 upgrade.

Cytrix was basically a two-tier client server technology. Through Cytrix, users could connect to the server through a user interface. “It automatically routes you to one of these servers based on workload and you can run applications from a very powerful server instead of running them on your workstation,” explained the operations manager. Because the application was run on the server, the logistic group was able to take the problem-causing application off users’ computers. “You can basically have almost a dumb workstation,” said the system administrator who championed the Cytrix project.

The initial problem of application incompatibility turned out to be a blessing for the logistic group. Following the discovery of Cytrix, IT personnel managed eight servers instead of hundreds of workstations across nation. “The benefit for us is it takes the administrative nightmare out of maintaining desktop PCs…. So now we don’t have to worry about the local desktop at all,” said the champion.

The project was so successful that Infrastructure Support Group who is responsible for quarterly SAP client upgrade decided to adopt the technology and move the application over to the Cytrix farm, dramatically cutting down the time needed to perform maintenance. “Therefore, we have eight servers to upgrade because that’s how many are in our farm instead of 1000 workstations.”

CHAPTER 6

CASE NUMBER TWO: SAP UPGRADE AT CP

6.1 Background

The second case study of this research focuses on the SAP upgrade from version 3.0F to 4.6C. SAP was a critical application running key business processes at CP. Most of its users were employees in business departments like accounting, finance, order management, scheduling and so on. At the time of the study, all business units at CP were on SAP. The centrally managed system could be accessed by all 44 field offices and manufacturing facilities through remote login to servers located at division headquarters.

The SAP system was managed by SAP Support and Development Department (SAPSD). Led by Angela, the department was responsible for the entire life cycle of the SAP systems, from mapping business requirements, configuring software, testing, implementing, to supporting business users. At the time of the study, the department had over 65 full-time employees and contractors. Many of them fell into one of two categories: functional or technical. Functional personnel were IT employees who had good SAP knowledge and extensive understanding of business processes. Each functional person belonged to one business area and worked closely with the users in that community. Conversely, technical personnel, although divided into teams, were shared resources. They were fluent in a “native” code, called ABAP, developed by SAP and focused most of their time on solving technical issues for the entire SAP department. Almost every SAP project was collaboration between the two groups of IT personnel.

While SAPSD managed the system at the application level, it relied on two other IT groups to provide technical support for back end operations. These two groups were UNIX

Operating Systems group and Database group. UNIX Operating Systems group managed the UNIX server that SAP ran on, whereas the Database group administered the Oracle database that was part of SAP system. The director of three groups met weekly to discuss and coordinate the SAP department's technical need.

6.1.1 Keeping up with SAP

Even though SAP was a mission-critical application, it was not standard software adopted by all divisions. Thus, no dedicated corporate task force was setup to assist SAPSD in the evaluation of new releases. The task of keeping up with software changes fell on the shoulders of SAPSD. Different venues were used to keep the department informed of on-going developments. "We are exposed to it through technical user groups kind of meetings, conferences with SAP and other businesses using SAP," said Yuri, a team leader.

There were two major conferences that SAP personnel attended. One was hosted by Americas' SAP Users' Group (ASUG); another was hosted by SAP itself, called SAPPHIRE. From the project manager's perspective, "ASUG, is really the customers'; it's really their organization." It was one conference that allowed different customers to network, to share their problems and also to voice their requests to the vendor. On the other hand, SAPPHIRE was described as "An annual conference that SAP puts on. It's really more of a sales presentation type. Here is all the bells and whistles, kind of thing. It's like a sale, lots of nice neat things to do. We'll send you to Disney World, you know, have a lot of nice receptions for you." The manager explained that SAPHIRE was a conference more targeted to attract new customers. As CP became a mature customer, they attended those conferences less compared to the early days of their adoption.

SAP assigned an account manager to work with CP's SAPSD director on an individual basis, informing her of any new development. SAP was also aware of the version that its customers were using. According to Angela, "Basically, they know what release you're on so they'll send you patches for your current release and then any new software at higher releases they'll send you because they try to encourage you to upgrade."

Besides regular updates, SAP also conducted Web casts and sent announcement to its users. IT personnel at CP who were on the conference mailing list received e-mail updates on software news. "They certainly over communicate," said the director.

6.2 Decision Making Process

The decision to upgrade SAP software was usually initiated by the SAP department and made by top management. IT personnel often are not involved in the decision making process. "Usually …we're told we're going to upgrade and the next thing, really, the timeline is put out in front of us," said Kellie, another team leader. As the director, Angela had all teams reporting to her and was in the best position to make informed decisions. The upgrade process usually began with her: "I typically recommend the upgrade but we have a business steering committee, basically the head of each department, but I'll make the recommendation to whatever project that we're going to do and they'll have to say yes or no."

The recommendation to the business steering committee was usually presented to a group of vice presidents from different business department, e.g., accounting, customer service, and production planning. To rally support for the upgrade, Angela usually met with each of the members individually before her presentation to the committee to convey the purpose of the

upgrade. “There’s a lot of salesmanship that goes into it. You can’t just walk into a big meeting once and say we’re going to upgrade, because they won’t go for it,” she explained.

Before SAPSD could make recommendations to the business departments, it first sought approval from its affiliated technical groups, Database and UNIX Operating Systems, to commit their resources for the upgrade. The process was described as an “easier sell” because IT groups usually “don’t want to hold back another IT group.”

Business units rarely objected to upgrade decisions. One business director explained that business units were not familiar with the direction of the software, and often relied on Angela’s group to inform them. “The reason being we don't know SAP, what they migrate to. We don't even know what version is out there because that's their area of expertise.” However, business units did negotiate to avoid implementing upgrades during busy periods of the year, like year-end, “It's more a negotiation around when, not if,” said the director of the SAP department.

While the business committee had to approve the date for software migration, SAPSD’s recommendation to upgrade was often guided by many factors. The next two sections will touch on both general policy and specific influences that led to the upgrade decision.

6.2.1 General Policy on SAP Upgrade

The decision to upgrade was usually guided by a few policies set at corporate and departmental level. Each of these policies was created with a different purpose in mind. The first policy is referred to as “no beta version.” This was a policy created to safeguard the department from adopting unstable software. The second policy was “18 months rule,” which was more of the director’s managerial preference than a strict guideline for upgrade timelines. The third

policy was “vendor support,” a company policy that required all packaged software to be on vendor support at all times.

6.2.1.1 No Beta Version

When it came to adopting new software releases, a cautionary approach was used to protect CP from implementing unstable software. The department had a policy prohibiting any upgrade of software that was earlier than release “c” in each version. “Like they'll have 4.6 a, b, and c. We never go before c, because a is the beta, b is the fixes, and c is the stable release. …Particularly, we don’t want a beta version,” the director explained with a grin.

In fact, their preference was to wait for other companies of compatible size to perform the upgrade first, “We want to make sure they have it in production and that other customers are using it before we consider using it.” There were two reasons for that practice: to sort out any known problem in advance, and to learn from another company’s upgrade lesson.

6.2.1.2 18 months Rule

Like many software users, the SAP department did not upgrade to every new version of SAP released by the vendor. “We don’t try to upgrade on a yearly basis, that’s for sure,” said Kellie, a technical leader of SAP group. Ideally, SAP department wished to conduct an upgrade every 18 months, “Because it takes about six months to do the upgrade, so that gives us a year between projects,” the director explained.

However, the SAP department had difficulty keeping up with the ideal timeline in recent years. SAPSD had conducted two upgrades between December 1996 and December 2000. First

upgrade, from version 3.0C to 3.0F, was carried out in October 1997, ten months after the initial SAP implementation. The second upgrade, which is the subject of this study, was conducted in December of 2000, three years after the first upgrade. One factor that influenced the SAP department's ability to follow the ideal schedule was the availability of IT resources, which are discussed in more detail in the SAP 4.6C upgrade section.

6.2.1.3 Vendor Sunset Date

A third factor that influenced the SAP department's timing to upgrade was the vendor's sunset date. According to the director, "We do want to always be on support. That is the company's standard that we will not run software that is not supported by the vendor." As a billion dollars business, CP's corporate policy required all packaged software to be on vendor support to ensure continuous system operation and timely recipient of vendor support if a problem occurs. In fact, the sunset date, when vendor maintenance ended, was one of the driving forces on the 4.6C upgrade.

6.2.2 SAP 4.6C Upgrade

4.6C upgrade was mainly driven by two factors, vendor's sunset date and new functionality in the software. The director explained, "That project was really two big reasons: one is we were off support. Two is we really wanted to implement some functionality that is just not in the older releases."

A similar account was provided by the project manager, Sharon, "SAP told you that we are not going to support you after a certain date. That was part of it. That was a big part of it. The

other part was people in the group and the business started to see SAP has really added a lot of functionality."

At the time they began planning in July of 2000, they had remained on 3.0F for almost three years. SAPSD was lagging behind on their upgrade schedule and were very close to vendor's sunset date. "We were losing support from SAP on our current release, they only support it for three years and we were behind," said the director.

According to the initial sunset date set in September 2000, the SAP department would have gone off vendor's support midway during their implementation. SAPSD began planning in July of 2000 and didn't complete the migration until December of 2000. Nevertheless, many organizations were still using version 3.0F and unable to meet the migration deadline. SAP ended up extending the support after they finished the implementation. Because of that, CP never really got off vendor support.

According to the project manager, vendor support was the key reason for the 4.6C upgrade. The upgrade was not really a choice for SAPSD; it was an ultimatum from the vendor. A functional team lead, Rose, agreed, "We were basically up on our contract with SAP. So, we were told we needed to upgrade."

One of the reasons SAPSD waited so long to upgrade was that IT personnel at the department had been pre-occupied with different SAP projects as a result of business changes. In the several years leading up to 4.6C upgrade, CP had been in an acquisition mode. As a result, IT personnel had been busy meeting the business needs, implementing one SAP project after another.

When SAP was first implemented in 1996, the systems were installed in one business unit hereby referred as the Consumer group. Immediately after the first upgrade, CP made a

decision to implement SAP systems in another business unit hereby referred as the Commercial group. The Commercial group had very different business practices, and the implementation was a massive undertaking. According to a functional team lead, “It was a much bigger division, and it was going to require a tremendous amount of effort to work with that division to understand business practices, business requirements, design, prototype development -- literally a new implementation.”

The Commercial group implementation turned into a huge project for SAPSD, consuming two years of their time. Because SAP was designed to be an integrated system, IT personnel had to add great amount of new functionality for the Commercial group and modify the existing system to accommodate both business groups. The department re-implemented SAP to the Consumer group in July 1999 and introduced SAP to the Commercial group in October of the same year. Immediately after that, they performed another implementation in an acquired business unit outside of headquarters. Meantime, an ambitious acquisition of a company twice their size was in process. Rose, the functional leader recalled that CP was going through tremendous changes during that period of time, and all of their attention was poured into meeting business needs. “We went from implementing a division to buying another company, so we ended up with another implementation, and then we ended up buying [NAC], which is a huge company. …so we had to put our upgrade aside so we were able to make those changes.” All of the competing projects caused the routine SAP upgrade to take a back seat.

Although they were many versions behind the latest 4.6C at the time, SAP 3.0F was described as a very stable software. After they had used it for many years, most of the known issues on the system had been identified and resolved. The director explained that their reliance on the vendor has subsided tremendously. “We had been live on that for several years so we

didn't need a great deal of help from SAP at that point. …Yeah, the longer you've been on a release, your reliance on the vendor becomes less so your incentive for an upgrade actually becomes less," said Angela

Nevertheless, vendor support remained important because SAPSD relied heavily on SAP Online Support System (OSS) to solve their technical problems. OSS was a database that contained SAP native code solutions that could be downloaded and applied to problems. When IT personnel encountered problems, OSS was the first place they looked for a solution. IT personnel logged in through a personal I.D. and either created a case documenting the issues at hand or retrieved OSS notes.

Because of their dependence on the help system, it was important that SAPSD stay current with vendor support. Even though SAP continued to provide technical support after the sunset date, it was done at an additional charge. "You can receive support from them, but you have to pay for it and they also treat you with a lower priority than customers that they have a support agreement with," said Tina.

One IT person had a similar observation, "We found that SAP was more responsive if you were on a newer release." She explained, "SAP wasn't going to continue to carry the overhead and the knowledge base to keep old people around for a very old release like 3.0."

Unfortunately, because of the competing projects, SAPSD could not upgrade to 4.6C sooner. By the time upgrade was planned, they were many versions behind the latest release. As CP fell behind the upgrade schedule, it missed out on two benefits that new software offered: new functionalities and fixes to bugs in existing software.

We had the upgrade for a couple of reasons. One is some of the bugs in SAP weren't going away and when we called SAP or sent an OSS note to say 'this is not acting right' and they'll say 'yeah, you're right. All of our customers complained and we fixed it in release, you know 3.1.' And, you're still on release 3.0f so you need to get off that old version of SAP and move to a more advanced one. We found out that we were getting further and further behind

- Tina, Functional Team Leader

In addition, as CP experienced huge business growth in the late 1990s, SAPSD began to feel that the software was incapable of delivering some of the functionality that the business needed. "It was working okay for us, and then we started to acquire more businesses and we were growing. The flexibility that we want to have on our software was not there in the older releases. The newer releases were so much better than the old one so," said the Director. One of the team leaders of a functional team recalled that functionality in 4.6C was one of the major reasons 4.6C was adopted. "So, a lot of different reasons; but the bottom line is, we need to upgrade because we needed a lot of functionality. We need a lot more functionality, a lot more flexibility with the system."

Upgrading to new version not only allowed IT to receive new functionality in the software, it also reduced the need for software customization. From IT personnel perspective, that was an incentive to upgrade. "Well, if we upgrade there, we can get all these extra added benefits. We don't have custom code, we just use what SAP has." said Kellie, a team leader.

As packaged software, SAP was developed with general functionality targeted to a specific customer base. Although complex and filled with functionality, the software was not

custom built to meet the business requirements of every company. From time to time, business units submitted requests for additional functionality or changes to the existing SAP systems. When that happened, SAPSD looked at existing software to determine if the features had already been built into the system. The director explained they used about 50 percent of the features. If the feature existed but was simply not enabled, the preferred approach was to configure the software by turning on those functionalities. If a request could not be granted through configuration, an alternative was to implement a “user exit,” which was a “placeholder” pre-allocated by the vendor for SAP’s ABAP code. SAP clients utilized user exits to implement some of the functionality that they needed for their businesses. In very rare circumstances, a modification to original SAP code, referred by IT personnel as “core mod,” was conducted.

As long as the request for functionality was built into the system, business users did not care SAP personnel did it. As the director explained, “So, the business doesn't care, they just want the functionality; they don't care whether we program it or it is in the software. But for the support perspective, we care, because the vendor takes care of it if it is in the software.”

Thus, being able to use SAP code rather than having to develop was something that IT strived for. “If it’s something that we’ve had to program and make huge efforts to get done in 3.0, where here it is sitting right in 4.6 already there for you. We prefer to not have to do a lot of custom coding. So if it’s something that SAP is going to provide then that’s a definite factor in making the decision to go ahead with the upgrade,” said another tech lead. At the time of data collection, the SAP department was combining the SAP system between CP and the NAC and had no plan to perform any upgrades in near future.

6.3 4.6C Deployment

The 4.6C upgrade was led by Sharon, a cheerful and enthusiastic eighteen-year veteran at CP. As a project leader, Sharon played many roles: she managed the overall project, ensured the project timelines were met, devised the final rollover plan, published the newsletters, and hosted events to motivate IT personnel. The upgrade was a huge project and lots of preparations were undertaken to make sure the upgrade worked. "The people who are in our group, have to spend quite a bit of time just to be able to replicate the process in the new upgraded version," said Sharon.

6.3.1 IT Personnel Training

To prepare for the upgrade, IT personnel started by reading release notes and attending training classes. Training was considered essential to provide SAP personnel with the knowledge of changes in the new version.

> *Because we are the ones who have to do the development and the configuration based on users' requirements, we need to know before hand what are these new features that are coming in the new release, what are things that may be going away. So training kind of helps that because you get to ask the instructors, "Will we still be able to do this in the same way or do you have to do it another way?"*
>
> – Kellie, Technical Team leader

The training was usually conducted at CP by a trainer brought in from SAP. The type of training that IT personnel received was referred as "delta" training, because IT personnel did not

receive a complete training on the entire software but the focus was put on introducing changes between previous and new versions. During 4.6C training, Yuri, a technical team leader, recalled that SAP had a difficult time finding a trainer who was familiar with both 3.0F and 4.6C.

> *We actually waited a long time between our upgrades. Typically their training classes are from the most current version that they've released, to one or two versions back. It doesn't typically include everything from the beginning of time. So, it was a challenge for SAP to find a trainer that was familiar with 3.0f and all the things that had been changed in the system up to 4.6c. So, the courseware that they provided didn't include everything that was changed from 3.0f to 4.6, for instance. So, in fact, I think it was 4.0, I think our training, the courseware was from 4.0 to 4.6c. So anything older than 4.0 it was up to the instructor to remember things that had changed between 3.0f and 4.0 and tell us about it.*

Even though the training was something of a compromise, Yuri was satisfied with the outcome. "It was a good delta training, I would say it covered 95 percent of the things that we needed to know."

6.3.2 Issues of Upgrade

Ideally, an upgrade should provide improvements over older version of software but at the same time preserve existing functionality. Unfortunately, SAP upgrades were never a quick plug and play. One reason SAP upgrades were labor intensive is software customization. Although SAPSD had a policy discouraging customization and prohibiting core modifications ("mods") unless they were absolutely necessary, customization to SAP was deemed inevitable.

From time to time, business functionality was added and codes were modified to accommodate CP's business needs.

Customization complicated the upgrade because modifications were being made to the underlying logic of the packaged software without the vendors' knowledge and would not be preserved in subsequent versions. During the SAP upgrade, core mods were overwritten and either replaced by new functionality that SAP put in or returned to its original state. Functionality added through user exits were marked by a special convention, as the project leader explained: "The convention is, if you have a custom object, you start them with a z and y. SAP knows not to use any of those; hopefully, they won't overwrite them." Nevertheless, according to the director, only 75 percent of codes were kept intact, and 25 percent were changed during the upgrade.

The internal logic of the software also changed from one version to the other. "Because they do, they change table structures, they change the data flows from module to module," said one IT personnel. In addition, a tech leader, Kellie, recalled that SAP was not very consistent with their naming convention: "The same function may be named one thing on the order management side and they may name it something else on the materials management side." Because of the uncertainty of the upgrade, each upgrade project was carefully planned and executed to make sure no adverse effect arose from the upgrade.

6.3.3 Upgrade Process

A typical upgrade took between four to six months to implement and each upgrade consisted of "half analysis and half hands-on implementation." According to Angela, "We spend basically three months going through all those documents in developing our detail plans. And then the last three months is making the codes, configuration changes, testing, and training."

The first step in an upgrade was to identify a list of objects that changed as a result of the upgrade. To ease the transition, SAP provided a reporting tool that “points out where they think you are going to have problems.”

The responsibility for retrieving the error report fell on the Database group. As the basic team in charge of SAP infrastructure, the Database group worked with the UNIX Operating Systems group to prepare the upgrade environment for the SAPSD Department. At CP, a new upgrade was never implemented directly to the production systems. Rather, a process called “Promote to Production” divided the implementation into three stages: development, testing, and production, with each phase done on its own servers. This approach allowed developers to make new changes on one server while business users performed tests on the other servers.

To run the reporting tools, the Database group would first install the upgrade on a copy of the production system in a development environment and perform database changes to transfer data from the production system to the development system. Then, using the reporting tool, “SAP will kick out a report of all the different errors that have been encountered due to the upgrade,” said Yuri, the technical team leader.

The 4.6C upgrade error report returned more than 1,100 objects that had been changed. Most of the errors pertained to core modifications or user exits. Sharon, the project manager recalled, “The Technical group had a large effort to go through and correct programs and change programs based on database changes.” However, she was glad that the error report was available to guide them. “At least they have a starting document that SAP generated automatically instead of some of the objects that you might have issues with.”

Once the report was generated, it was passed to the technical team for the initial phase of programming changes. Technical team members went through the list and fixed any reported technical issues in the system before passing the system to the functional team for thorough testing. “We reapply any core mod and user exits and at some point the functional team will start executing their test plans,” said Kellie.

The upgrade process at CP was a shared responsibility between technical and functional teams. Technical people were the forerunners in the SAP upgrade, making technical corrections on reported issues before a full-scale system test could go underway. Functional people were described as “leaders in testing,” performing the majority if not all of business function tests, in addition to configuring systems and training business users.

6.3.4 Upgrade Policy – No New Functionality

At CP, one of the most important goals of an upgrade was to preserve existing functions that the business used. SAPSD’s upgrade policy was to focus on getting the new version of SAP working with all existing functionality and not to introduce any new functionality during the migration.

There was a reason for such a careful approach. SAP was a critical system running daily business operations, and SAPSD would not risk any chance of it not working properly after the upgrade. One IT personnel explained, “SAP is used for us to externally communicate to the customers. So, we really had to focus on testing because we couldn’t afford the downtime of not communicating with customers. We couldn’t afford for customer service not to be able to enter a sales order or for one of our carriers not to be able to pick up a shipment and take to Walmart because the system didn’t work right.”

Even though error reports were available to assist SAPSD, it was merely a starting document that IT personnel used to solve issues detected by the reporting tool. In the past, the SAP vendor had made changes that were undetected by the reporting tool and were not discovered until a much later time.

To preempt undesired system failure, SAP adopted an extensive testing strategy to weed out any potential software issue. According Kellie, “That’s why we have to make sure we really test, especially the business process. That’s why we will create an order, create a delivery and, create an invoice. That’s why we have to go through the whole business process as if we were a user actually doing day-to-day business processes. That’s why we run the batch job and we review them. If they fail then we get on them immediately to see why did the job fail.” The goal was to make sure all business functionalities remained intact as a result of the upgrade, “because at the end of the day, we wanted the same results they were already getting,” said functional team leader, Tina.

6.3.5 Testing

As soon as the technical team finished its initial code changes, the system was passed to the functional team to begin testing. Each functional team was responsible for testing a sub-area of the business process. “The first thing we did was we divided the areas that were impacted amongst our team members. So we had somebody working on the accounts payable, somebody working on the accounts receivable part, somebody working on general ledger part,” according to a functional team member who was responsible for the financial module.

A “test plan” testing strategy was adopted to assist functional people perform checks on the system. “We have a very structured testing methodology. We have approximately 1,100 to

1,200 test plans," said the project manager. A test plan was like a blueprint of how things were done in the previous version. It detailed steps needed to process a business function in SAP, along with the test data and expected outcome if the steps are followed through. Executing test plans was usually the first step in the testing process. The goal of testing was to identify any changes that had occurred in the new version that could affect the business process.

> *So if we executed that functional test plan and it didn't yield the same results, we knew that there was something wrong. Either something changed with the release that we had not taken into account of in our test plan or the way we did that particular thing in the system is now obsolete and SAP says 'I'm on a newer release, don't do it that way anymore.'*
>
> - Tina, Functional Team Leader

Each test plan usually coincided with one business process. Even though not all test plans got executed, the majority did. A business area usually had many test plans, "and even one of the test plans alone is 70 pages. It's unbelievably huge," said Rose. IT personnel unanimously agreed that testing was the least "fun" step in the entire upgrade. "Testing is an awful period … writing the code is fun, doing implementation for business is fun, but that middle piece of testing is not fun," said the director.

Because testing was such a tedious task, the project manager would organize "fun activities" during the upgrade period to maintain a degree of enthusiasm within the team members. "It can be very painful to have to execute 25 test plans, which may be 50 pages each. So, we do a lot of fun things within the group, like we had Wacky Wednesday. We would have a

different theme each week like wear you ugliest tie. We actually had contest among different team members and we had prizes, so we do some fun things to get people going." She recalled that those activities were really fun and was grateful to the director who gave them a nice budget to host those events.

6.3.6 Issues Resolution

During the test, if the results of the test plan were found to vary from anticipated results, IT would do research to see why something was different. One approach used was to survey OSS notes to see if the problem had been reported and if any known solution existed. In the situation where solutions were not readily retrievable from OSS, IT personnel submitted an OSS message to SAP. "For the most part I've gotten good response from them, but you do hear people say 'Well, SAP hasn't responded yet' and it may have been two or three days so you try to check up and see where they are; and if you're not getting any responses, we do escalation," Kellie recalled.

The escalation process involved contacting a vendor representative "We have one person that was assigned as our contact, if we were not getting timely enough response from SAP, he would intervene," said the project leader. Although SAP was fairly responsive in solving most of the problems that SAPSD submitted, the director recalled some nerve-wracking experiences in which the vendor waited until the last minute to resolve their problems. "We find bugs in their software and we have to convince them that there is a bug and they have to fix it in Germany and get back to us. And that can be back and forth, back and forth trying to convince them that it's them. … They will not believe a lot of time that their code … it can't possibly be their code. And we are sure it's their code (laugh)." The director explained that the situation was usually "very

tense right up until then." Often, SAP waited until the last minute to give them the code. "And we are on a timeline and we want to upgrade on a certain day and they have 500 customers, and they just treated us like another customer. So honestly that is our biggest problem," said Sharon, the project leader.

Although SAPSD eventually received the fix, as a fall back, SAPSD always developed contingency solutions in case SAP did not provide the solution. "On our side, we spend a lot of time looking at if we have to do it ourselves. What are we going to do? We have a contingency plan. Honestly, what starts stretching the upgrade out is this going back and forth, developing this contingency, testing this contingency. … It probably added a few weeks into our project plan," said the director.

To the director, one of the solutions to the problem was vendor management, getting the vendor to see their need and often proving to the vendor that the demand was legitimate. In addition, they always built in a 10 percent additional timeline in their project plan to take into account the vendor's potential untimely response.

6.3.7 Training

While testing ensured that software functioned properly, training was done to make sure that users knew how to use the software after the upgrade. The 4.6C upgrade brought a huge change in user interface, "We went from 3.0fF to 4.6C and that was like thirteen releases at one time, so that was a very big change. And when the screen changes, we need to re-train them," said the director. Users were given delta training during the 4.6C upgrade. According to a Functional Team Leader, Tina:

We're not going to train on complete functionality. We're only going to train you on the changes. So, what we did was say, "Business processes are the same except you know you used to click on the top of the page, now you have to click on left of the page and it will give you the same thing."

All of the training was held at the on-site computer labs. During my visit, I attended a training session for a business function change. Even though it was not 4.6C training, it gave me a good idea of how the training was conducted. Training session often combined lectures and hands-on exercises led by one or two instructors. Often, users were given a manual and the instructor went through pre-designed training course to familiarize users with the upgrade and highlight the changes. Throughout the training session, baskets of candies and chocolates were available on user's desk to spice up the long training day, and small gadgets were distributed to participants at the end of the training session to commend their effort.

During the 4.6C upgrade, a "train the trainer" approach was used to conduct the training. "We sat down and trained the power users so they could assist us. But, we developed all of the materials that they used," said a functional team lead. This approach was used to lessen the strain on IT resources. A business manager recalled, "[IT] didn't have any resources to train and we had to become the trainers." The training was usually held a few weeks earlier before the formal training class begins. This allowed the trainers to familiarize themselves with training materials and to become fluent with the upgraded system.

Even though formal training was provided, business managers realized that a few days of training was not enough to enhance the learning experience. CP also had a computer lab where users could go to practice. "Not only do they sit in the class and learn along with an instructor,

there's lab that you can go to after you have this training to practice what you learned. And, it's usually an instructor available in those labs," said Ella, a business team leader.

Besides providing training to users at headquarters, SAPSD also trained users in field offices and manufacturing facilities. The project manager recalled that this was sometimes a challenge, especially when the facilities were a "union shop." The project manager also explained it was difficult trying to find a trainer who can train at odd hours.

> *So, there has to be an agreement with the union, you know, that you can't take them away from their regular work hour, they have to either come in early or stay late. … And trying to have trainer who can train at midnight, you got to have training class from midnight to four o'clock in the morning, to train the people.*
>
> - Sharon, Project Leader

6.3.8 Role of Communication

Another key issue during the 4.6C implementation was to make sure the timeline of upgrade was followed. One strategy that Sharon adopted was tracking the progress closely and communicating the status fervently to all IT personnel. Approximately two months before the final cutover, she reported each team's progress on testing and development during daily meetings. "I would be staying on top of people, I would send them notes, you know, your percentage complete is only at this level. Being on this level, you won't meet your date. It's amazing, the next time they reported, their percentage complete would be closer to what it should be. But tracking it at a pretty detail level and communicating with people, I think are very important." Sharon felt that constant communication of urgency was one of the keys to success.

6.3.9 Internal Audit

Before SAP went live, it first went through an audit process. At CP, any upgrade or change to applications with implications for GP's financial reporting must have received approval from the audit team before it could be used. According to the director, there were only four to five applications at that divisional level that fit into that criterion.

The audit team looked at the documentation and methodology that SAPSD used to prepare for the upgrade. In addition, they interviewed business communities who were involved in the testing process. Audit was meant to "guarantee the quality of upgrade," said the director. "Ultimately, we don't want to impact our customer."

The project manager worked closely with the audit team to satisfy their inquiries. "I was responsible for working with internal audit, to make them comfortable that we had all the mechanisms in place and made sure that everything was thoroughly tested, that we weren't going to impact the business." Furthermore, business users signed off on the testing. The audit team wanted to make sure that all parties affected were aware of the major changes in the upgrade.

However, the degree of auditing was proportional to the complexity of the upgrade. During the 4.6C upgrade, they met regularly with the audit team to "go over everything," according to the project manager. "They want to make sure who you have involved. Who are the businesses who've signed off? What are your testing procedures? …To make sure that everything is signed off." Basically, the audit team analyzed the overall situation to ensure steps were taken to reduce the implication of upgrade. "Ultimately, we need to get a low risk rating in order to go live," commented the director.

6.3.10 Cutover

At CP, the actual upgrade of SAP into the production system was referred as “cutover” or “systems go live.” Months of testing and development were meant to prepare for the final system migration into production. Going live was a huge event because SAP was not only a complex system but also a business critical system.

To prepare for the cutover weekend, the project leader prepared a detailed implementation plan. The plan consisted of “steps that need to take place over that weekend, because there are things that have to happen in a certain sequence,” according to Yuri. Before the actual cutover, SAPSD and the database team conducted a mock upgrade to make sure the upgrade would go smoothly. According to the director, two to three trials were common because there was no room for error and they received no forgiveness if things went wrong.

The cutover date was pre-negotiated with the business community and the upgrade was held over a weekend. At CP, some systems operated on a 24/7 basis, so shutting them down on a weekend could lessen some of the impact.

The process usually begins on Friday evening. Before the system was shutdown, functional team member would print reports for system validation purpose. “We printed off a trial balance; we ran cost center reports to say here’s what the balances ended in 3.Of when they bring up 4.6c, the balances and all the transactions should be there.” During the migration, functional personnel would perform a quick test to validate if the implementation of certain functionality has gone well and lots of checkpoints were put along the process to assess status of the upgrade.

The Database team began the migration process. They installed the new version of SAP on the production server and migrated data from the old system to the new. Technical and

functional personnel then put in the changes that needed to be applied to the upgraded version. The cut over weekend was intense. “That's when you stayed awake for 72 hours, you know; you only go to sleep for a couple of hours at a time,” said the project leader.

Because all steps had to be executed in sequence and according to plan, communication became extremely important to inform personnel of the status of the upgrade. A special phone line was set up to report the status of the upgrade, “We had a hotline, that we kept current the status of the upgrade. ‘We are currently two hours ahead; we are currently six hours behind.’” IT personnel called in to find out if they had to come in during a particular time. In addition, rooms in nearby hotels were reserved for those who needed to stand by. As intense as it was, the upgrade was completed in one weekend. “It was just a weekend. We started on a Friday night and by Sunday everything was back to normal,” said Tina.

6.3.11 Post Implementation Support

Immediately following the cutover, members of the functional team provided technical support to help business users adapt to the upgrade. Personnel from functional team “walked the floor,” making physical appearances in business departments to answer questions from users. A business user, Jane, explained: “If we do it over the weekend, that Monday they come in and walk the floor to see if everything is okay. …and they usually do that like the first maybe two or three days.”

In the first two weeks, each floor was assigned to a functional member. If users had any question, the assigned functional member assisted them. The SAP help desk was also available for user to call at all times. “We have an SAP helpdesk, so what we did was, if there was any

issue, whatever, someone got stuck and couldn't figure it out, we can call the helpdesk," said Felicity, a business manager.

6.3.12 Business Approach to Upgrade

Business users took the SAP upgrade seriously. Although the main execution of SAP upgrade was left to SAPSD, business users adopted an active role in the preparation process. According to Felicity, "We tried to do a lot of training, a lot of preparation ahead of time. So, we were heavily involved." A business team leader, Ella, explained they were willing to participate in the upgrade process because the SAP system was critical to them.

> *And most people want to be apart of those kind of things because they feel like they use it everyday, and it's important that they participate and have some feedback. Otherwise we could end up with a Frankenstein and no one can use it. So they understand that they have an interest in the final output towards whatever is created. …There's a major willingness to participate in things like that.*

One of the preparations was testing. Jane, a subject expert, recalled her participation in pilot team: "They would get a group of us together and we'd go down and we'd play and say, 'What about this …What about the export orders?' …Basically to see…if we ran across anything that we did in the old system or the old version that new version didn't capture." Subject experts entered orders into SAP to see if there was any discrepancy between the old and new system.

Business not only participated in pilot testing, they also signed off on the upgrade. Ella recalled, “I’ve always been one to sign off on the design after we pass all the testing. I would sign off and say ‘okay, we’re okay with this new design.’”

Most importantly, to ensure that users were able to perform business operations competently after the upgrade, business management mandated all users to attend training. In addition to formal training, business users took advantage of labs. Business manager, Felicity, recalled: “We make sure we have lab where they can fool around with the system. We schedule lab time.” Training was even more important in departments that dealt with customers because “We don’t want our customer to suffer,” said the manager.

After the upgrade, if users ran into problems, they consulted each other or brought the issues up in their weekly meetings. The strategy of having subject experts also helped to make the transition easier. Subject experts were power users selected from each group and could more easily address user’s need.

> *That’s why we picked six people and that was basically one person from every team so that each team has someone sitting near them. They can just holler over “Hey, I’m stuck. Help me.” Or they could hear that person because some people won't tell you they're stuck if they have an issue. If you’re sitting with them you can hear them slamming the keyboard and go over there and say “Excuse me, what are you trying to do. Let me help you” or whatever.*
>
> - Ella, Business Team Leader

Although it was critical to management that users were able to perform their daily tasks without too much interference, managers also understood that the upgrade could change the system and put stress on user's emotions. According to Ella,

> *Most of them are outstanding performers or very good performers, and any time you make a change they have to pretty much start over and now they're operating again at a competent level or a very basic level. … And so when people are stressed like that their performance and their ability to focus is not where you want it to be. So you have to get them through that.*

One strategy that some business team leads used was to be understanding and provide motivation and reassurance if a user felt discouraged.

> *We would walk around, asked people questions, you know, reassured them that they were doing a great job. If we sense people are getting frustrated we sent them to lunch early (laugh). We asked people to take breaks. We've always encouraged them to take breaks. We brought…toys or gifts …little something, you know, candy ...we may bring snacks, or donuts, something for breakfast to get people energized. And, we may have popcorn, something in the afternoon throughout that whole entire week to let people know that the business does care. We're not trying to stress you out. We're not trying to kill you (laugh). But it's critical to the business that you learn a new process and so we try to do some things to keep people OK during the day while they're here.*
>
> – Ella, Business Team Leader

Another business manager, Monica, recalled bringing in breakfast and pizza during lunch the first week of implementation to motivate users, "…little incentives just to make them feel better about what they were doing. And a lot of it was just going around patting them on the back, 'Hey, I know your pain you're going through a lot. We appreciate it.'" Knowing users could be under pressure and feeling frustrated from time to time, business leaders adopted an understanding approach to encourage learning and assist users adapting to the new system.

6.4 Impacts of SAP Upgrade

6.4.1 Impacts of SAP Upgrade on Users

The 4.6C upgrade was a technical upgrade, done out of technical necessity and not to fulfill business requirements improvement. The business users did not expect to receive any new functionality or experience any change in their business processes. According to a business manager, Julie, "Did we get any more functionality? No, that wasn't the intent. The upgrade was primarily technical, it was not to improve functionality." To business, a technical upgrade usually brought minimal impact; however, the 4.6C upgrade was a slightly different case.

During the 4.6C upgrade, a completely different user interface was introduced. A business user, Mary, recalled, "The 4.6 upgrade …affected our screens and everything. It was almost like having to learn the system again." A power user, Teresa, concurred, "That's when all the screens completely changed and that was the first time that we did have to tab."

User interface changes were observed in three respects: screen layout, screen navigation, and screen information. Prior to the upgrade, the user interface was menu-based; the user

interface for 4.6C resembled Windows-based applications, with icons, scroll bars, and tabs displayed on the screen. One business team leader observed that, after the upgrade, screens were filled with “fancy icons” that they could just click instead of having to perform everything from the menu bar.

Another interface change was new “folder tabs” that allowed users to move into different screens. “It didn’t have these individual tabs here,” said Teresa pointing to the screen. “Everything was from the tool bar and you went to wherever you wanted to go from the toolbar.” Before the tab feature was introduced, the user moved from one screen to another by going to the menu bar and selected their next screen through a pull down menu. After the upgrade, the user could move in and out of different screens by clicking on tabs spread across the middle section of the screen.

The screens on 4.6C were also described to be having more information on them. “The upgrade has so much more to look at. ...The screen’s busier,” said Mary, a business user. “And we had more fields on the first screen than we ever had before. Most of the fields you can’t see them because they are all to the right and you can’t see because the screen is only so big (laugh),” said Ella. Another user, Felicity, found herself constantly scrolling up and down the screen.

To help manage screens, SAP included a new feature that allowed users to customize screens and eliminate any fields that they did not want to see. However, Ella explained, use of that feature was cautioned at the beginning of upgrade, “We were a little afraid that people may hide something they think is not important, but it really is important to business that you see that information.” Until users were familiar with all the changes, they were discouraged to customize their user interface.

Besides changes on user interface design, SAP also altered some of the transaction codes. One user, Teresa, recalled, “Transaction codes were different from screen to screen.” Transaction code is a short code that users can enter to arrive at certain processing screen directly, skipping intermediate mouse clicks or keystrokes. “Like if you could just type in the transaction code and go to a screen and it would take you there …instead of doing the long drop down and then go and go and then end up here.” When the transaction code changed, users had to adjust to it because the original codes had already been ingrained in them.

With all the changes that the SAP 4.6C upgrade brought, some users saw immediate improvement with the new user interface. “It was so much better,” said Ella. Other users held a more neutral attitude. According to Teresa, “There wasn’t any impact on the productivity negatively or positively… It was just a different way of doing things.” The change was a part of the upgrade that they had to adapt to.

6.4.1.1 Learning Curve

Even though users had gone through training, the actual learning usually occurred when users had to perform daily operations after the systems had been upgraded. According to business managers, it usually took business users approximately 90 days to get back to their normal level of operations. In fact, one of the biggest impacts of the upgrade was the user’s learning curve.

A business team leader, Ella, provided a glimpse into users’ work life shortly after system conversion: “People trying to remember ‘Did I learn that?’ Because you can have information overflow. ...Did we learn that in class? They’re looking in the book trying to find out where it’s in the book and asking one another. And then, of course, someone has it all wrong and gets

everybody all confused. And also we have resources -- we can call a number and ask someone at the help desk. You have your learning curve," she said.

The upgrade not only changed the way users processed a transaction on a screen, it also altered the flow of the process from one screen to another. According to Felicity, "The major thing was the flow, the way they had it set up. You know, like the …old order management system that we had; it kind of prompted you to go to the next screen. You complete one screen, you go to the next. In (upgraded) SAP, you got to know where to click to go to the next screen."

In addition, because of users' unfamiliarity with the change, "We may not get it out as fast or we may not get it right the first couple of times," said a team lead. Initial productivity drop was inevitable at initial stage of upgrade, but in the end the business team lead felt the change was beneficial. "After you get through all those initial 90 days …where you thought it was going to do one thing and it did something else or …people are on it longer cause they're stuck on a screen not knowing where to go…. When you get through all that, …the change has always been better," said the team leader.

Although users would complain at the initial stage of upgrade, almost all parties agreed that once they got to use it for a while, they became accustomed to the change. Felicity recalled, "You know what, I think everybody was afraid of it at first, and now it's gotten to be like second nature, just like any new system."

One director observed that change could be tougher on veteran users because of their comfort level in using the system. "They know all the bells and whistles and how to make the system work with them as opposed to someone that's a new recruit," she said.

6.4.2 Impacts of SAP Upgrade on IT

Unlike users, the impact that IT received from the business application upgrade was the time and effort poured into the execution of upgrade. The 4.6C upgrade was a huge project to IT. “It's much closer, to me, to a second implementation than what I would consider an upgrade,” the project leader recalled. It was the biggest upgrade attempt that SAP Support and Development has ever experienced. “We made a very very big show. We went from 3.0 F all the way to 4.6C,” said a functional team member, Rose, who also equated the upgrade to “almost a completely new implementation.”

Understandably, an upgrade is more difficult to implement if it is a multi-version upgrade. The upgrade from 3.0F to 4.6C was a “13- release” upgrade and took IT approximately six months to complete. The more versions skipped, the more changes were likely to be found in the new versions rendering the implementation process difficult. “So, like with 3.0f to 4.6c, that was such a big jump and the larger jump it is the harder the implementation of the upgrade is,” said Yuri, a technical team leader. “So, everything has a complete different look and feel, said Sharon. “I mean people will go into it and it’s like you get shell shock, you're like, I thought I knew how to create an order, but this is very different.” The screen changed so dramatically compared to version 3.0F that even IT personnel could not recognize it.

6.4.2.1 Training

The drastic change in the software user interface meant that IT personnel had to re-train users to use the systems. Training was not limited to business users; even IT personnel had to be trained on those changes: “We had to retrain the support team members because we had never seen the screen the way they operated …we knew how to put in the order in, created a delivery,

created a shipment, build it. We knew how to work standard reports and all of a sudden, everything changes," Sharon said.

Before functional personnel could train, they had to develop training materials. Although they hired a training consulting firm to develop the manual for users, IT personnel had to identify changes and worked with the consultant to create a manual of their vision. Performing training was not as simple either. Even though IT personnel adopted the "train the trainer" model, there was still a huge effort to deliver training to different business groups.

> *…The fact that you weren't limited to a small group of end users. …You had to train customer service, you had to train customer account reps, you had to retrain the pricing rep, you had to retrain all the individual at the manufacturing facilities, you had to train your finance people.*
>
> – Sharon, Project Leader

Another time-consuming process was testing. In fact, testing and training were considered to be two of the biggest impacts in upgrade preparation according to the project leader. "I would have to say the two biggest issues of upgrading, was, number one, the amount of testing effort to go through and … number two would be the training of the end user."

6.4.2.2 Testing

As discussed in Section 6.3.5, testing was the key strategy used by SAPSD to safeguard against potential problems. IT personnel poured a tremendous amount of their time into making sure upgrade would work without glitches. As a large division in a *Fortune* 100 company, CP

could not implement a problematic system. To minimize unexpected issues, SAPSD conducted in depth testing on the SAP system before the application was deployed division wide. The 4.6c upgrade was especially time consuming because of the many versions skipped between the two upgrades. “It was such a huge jump in versions, even the screens were completely different, which of course meant … we were going to have to literally test every transaction that we used in SAP. So that's going to be a tremendous amount of testing that was going to happen,” said Rose. Even though not all functionality was examined, the majority of business processes were tested to make sure the systems functioned properly after the upgrade.

6.4.2.3 Customization

Another condition that complicates upgrade is software customization. Yuri explained, “The more you customize your package software, the harder an upgrade becomes because the more you have to look at and change.” Two types of modifications were made, user exits and core modification. Although user exit was a pre-allocated space for users to put in customized code, only 75 percent of the changes in user exits were preserved. Thus, thorough testing was necessary to make sure those changes were not impacted. Furthermore, SAP also put in core modification in rare occasion. Because core modification is done to SAP programs, its changes often got overwritten and need to be re-applied during upgrade.

> *The negative thing about upgrading is that when you upgrade, SAP creates different codes …that core modification no longer is in that program. So, what does mean? That means that we have to put it in all over again and we have to hope that it works. That is one of the biggest challenges that we have when you’re upgrading, is um, you may loose*

that code. You may no longer have it when you upgrade. There were several instances where we actually had to rewrite the code all over again because we lost it when we upgraded …

- Rose, Functional Person

6.4.3 General Advantages and Disadvantages of SAP Upgrade

In general, the upgrade provided several advantages. To the director, adopting packaged software gave them competitive advantage because it allowed the organization to focus on business needs instead of developmental detail of the system. “We don't have to maintain the base product, like accounting; we know it works. So, we don't really have people focus on accounting, we have people focus on what's the new functionality the business needs.” The upgrade provided similar benefits allowing CP to receive new functionality from vendors and other businesses. The director explained that SAP worked with organizations to develop specific functionality. Often, the development cost was shared by both parties, and SAP waited a year to put the developed functionality into applications, making it a standard feature. By upgrading, CP received the code development that other business required, as implemented by the vendor.

The upgrade also allowed CP to receive code replacement for functionality that they implemented. In fact, that was one of the attractiveness for going to the next version. By using code implemented by SAP, IT did not have to continue maintaining the code from one version to the next. The vendor became owner for that particular functionality. Nevertheless, code relegation was a double-edged sword. While vendor owned the code and the responsibility to solve any issues, it also controlled the internal code changes. Changes in functionality, user interface, or programming were now all in the hands of the vendor.

One IT functional person, Rose, recalled that SAP changed the text screen making it more complicated to work with: “SAP completely revised that screen and it’s not user friendly.” She explained that the change puzzled her. Not only the screen was revamped for users, it also brought problems for SAP personnel. When they approached SAP to inquire, the response was to accept it. “SAP came back and said this is the way we’ve designed it, pretty much live with it… I could not understand why they did that.”

6.4.3.1 The Effect of Software Changes

The biggest change in the 4.6C upgrade was the user interface. When a screen changed, it not only affected the user’s learning process, but it also had implications on implementation. According to Rose, “Once you change the screen then you’re talking about major program change.”

One area that needed attention was the back-end process. To speed up transactions, IT had setup programs that would automatically enter data into a screen, bypassing human intervention. When a screen changed because fields had been added, deleted, or moved, all of those programs had to be re-coded. “In the last upgrade, selection screens changed in some of the standard SAP programs so that would cause a job to fail,” said a technical team leader. “So, when the screen changes that meant also we had to go back and change all those programs because now that’s a different flow to the screen,” said Yuri, the technical leader.

6.4.3.2 The Effect of Vendor's Delay in Providing Solutions

One of the issues that SAPSD faced when preparing for the upgrade was getting the vendor to provide software solutions. The director recalled they had "three big ones" that could have deterred them from meeting the cutover date. SAP did eventually fix the problems, but they waited "until the last minute." This was a problem for IT because they had to develop their own solution in case SAP never provided one, which affected the project timeline. After the upgrade, the director recalled, "We had no drop out of productivity or anything on the last upgrade." Although SAP did not survey users on the impact of upgrade, they kept a close eye on business measures to track the performance of the upgraded system. Business measures were statistics on system performance that the Database group collected daily on the SAP server. While there was no "show stopper," the director recalled they were some bugs that "quickly got fixed."

A technical team lead, Yuri, recalled a problem that was not caught during testing and they had to make some emergency fixes a few weeks after they went live. SAP had changed the way data were stored in the database, and IT personnel didn't catch all the places where the function module was called. "So users were complaining after we went live that all of a sudden certain areas weren't working anymore." The problem was pretty serious because users had gone through the whole order sequence not realizing that the order was changed. According to the team leader, "we had to go back and actually clean up orders that had already been created."

The director explained that they seldom had major problems on the upgrade, "Because we spent so much time on the upgrade. Some companies will just do it and they will have this dip. It's decided on your approach, you know. We've just decided that would not work here." Thus, the upgrade was both labor intensive and time consuming. Each upgrade took an average

of three to six months of IT time, and 4.6C upgrade, which was liked to be “second implementation,” took six months to complete.

CHAPTER 7

DISCUSSION

Section CHAPTER 5 and Section CHAPTER 6 presented the case results for each technology upgrade. Across both cases, similar results as well as different outcomes were observed in all three research areas, reflecting the influence of organizational contexts and technology characteristics. A careful cross-case analysis induced the following integrated model as shown in

Figure **4**. The model combines the results from all three research questions and consists of six components: decision, motivating forces, contingency forces, planned strategies, corrective actions, and impacts.

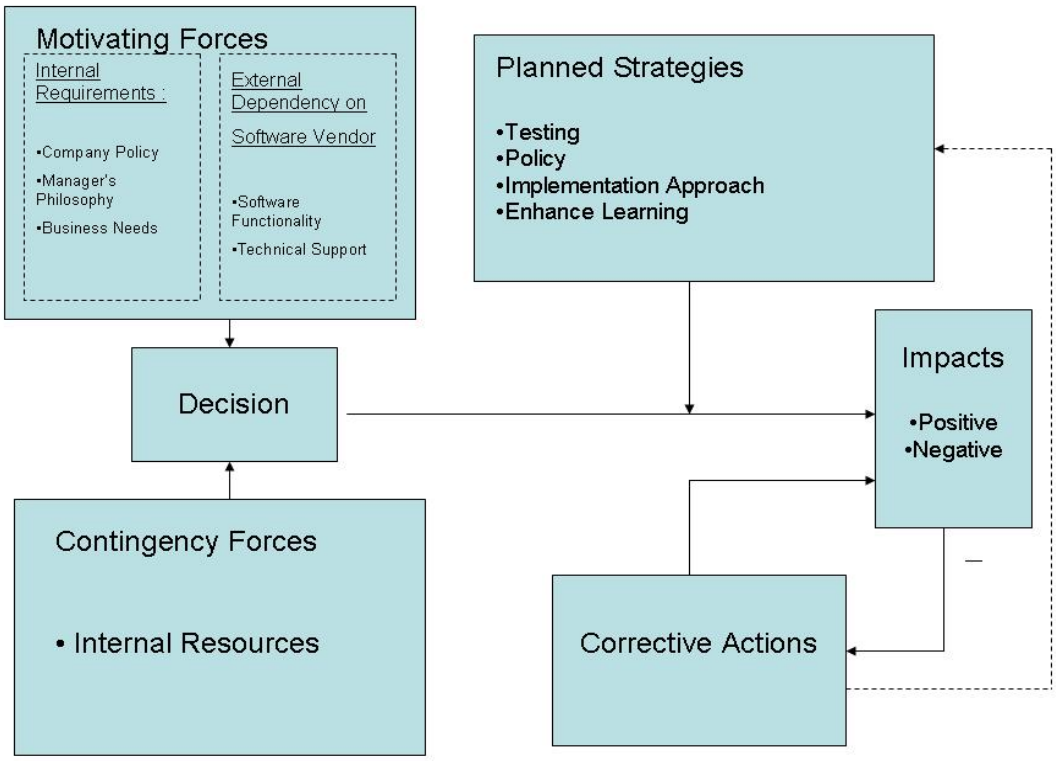

Figure 4 Induced Model

Two boxes, motivating influence and contingency influence, were drawn with arrows pointing to decision. The relationships of the components are such that the decision to upgrade is the outcome of interaction between motivating forces and contingency forces. Because a decision to upgrade will inevitably lead to positive and negative impacts as experienced by stakeholders, an arrow is drawn from decision to impacts. Besides the decision to upgrade and its impacts, two types of coping strategies were observed from the study. They are represented by the planned strategies and corrective actions components. The planned strategy is put in place to reduce negative impacts of upgrade, and corrective actions are solutions devised to react to negative impacts resulting from the upgrade. The presence of a planned strategy can mitigate the final impacts of an upgrade and is denoted by a moderating relationship with a vertical arrow drawn from the planned strategy to the line connecting decisions and impacts. To denote the reactive nature of corrective action, an arrow is drawn from impacts to corrective action. A negative sign is placed at the beginning of the arrow to show that the action is created to counter negative impacts. Because corrective action exists to remedy specific problems, an arrow is drawn from corrective actions to impacts, representing the changes in impacts that the corrective action incurs. Finally, an arrow is also drawn from corrective action to planned strategy to demonstrate the formalization of corrective action into a permanent planned strategy. However, not all corrective actions will become part of the planned strategy. Hence, a dotted line is used to demonstrate the optional relationship. The above description provides an overview of the model and its relationships; the following sections present a more thorough discussion of each component and its relationship with other components.

7.1 Packaged Software Upgrade Decision

Software upgrades, which have often been categorized as maintenance activities (Nah, Faja and Cata 2001), are very different from traditional maintenance that usually deals with systems developed in-house. When systems are built in-house, any changes to the system are performed by IT at their own discretion. However, with packaged software, most changes are produced by the vendor and distributed to users in the form of service packs or version upgrades. Modern packaged software upgrading is a crossover between traditional maintenance and new system implementation, with the exception that IT and users have worked with an older version of the software before.

Once packaged software is adopted, an upgrade is inevitable unless the organization decides to abandon its current software. According to Kremers and van Dissel (2000), companies that adopted packaged software "were of the opinion that migrations are an unavoidable part of the software life cycle. The question is not 'Should we migrate?' but 'When do we migrate?'"

In the exchange relationship between an organization and a software vendor, the parties have different interests. Ideally, vendors want organizations to upgrade frequently, thereby locking customers into their product and also reducing number of versions that they need to support in the marketplace (Kremers and van Dissel 2000). Organizations, on the other hand, feel that upgrades should fit into their technology management strategy. Most companies do not upgrade to every version that is released.

The decision to upgrade at CP was governed by multiple influences that can be classified into two categories: motivating forces and contingency forces. The final decision reflects the interaction between those two forces. The discussion will begin with motivating forces, which

can originate from both internal requirements and from the organization's dependency on the software vendor.

7.2 Motivating Forces

From the study, multiple motivating influences were found to influence the decision to upgrade both SAP 4.6C upgrade and Windows 2000. Motivating forces are defined as any event, policy, or requirement that stimulates the interest to adopt a newer version of packaged software. In this study, motivating forces originated from both internal requirements and the organization's dependency on the software vendor.

7.2.1 Internal Requirements

Three internal requirements were identified in the study to have created influences on the decision to upgrade. Two of these requirements, manager's philosophy and company policy, guide the upgrade timeline. The third, business need, is a dynamic requirement that changes over time. The following sections focus on each of these elements.

7.2.1.1 Manager's Philosophy

At CP, the decentralized structure put the decision power on the head of each department. According to Blankenship and Miles (1968), this is a common phenomenon in large companies where upper level managers "claim greater freedom from their superior" in decision making. Manager's philosophy provides a guideline on how aggressively they want to keep up with new software releases. SAPSD wanted to upgrade every 18 months, and the Infrastructure Group

wanted to keep up but not be overly aggressive. Overall, both groups adopted a moderate stance on new technology deployment, reflecting the organizational climate of a non-technology company (Swanson and Beath 1989). This comes as no surprise since IT departments are units within the corporation; the department's maintenance policy is usually affected by the company's climate for innovation. In both cases, the policy serves as guidance to manage technology changes and can be bent to make concessions when there are competing projects.

7.2.1.2 Company Policy

At CP, with millions of dollars at stake each day, the company had a strict policy that required all packaged software to stay on vendor support at all times. Although the policy remained dormant when software was not close to the end of the vendor's sunset date, it became a powerful influence when software approached the sunset date without being upgraded.

Another company policy was "no beta version," which both SAPSD and ISD observed. The policy governed when packaged software could be adopted, helping to prevent the adoption of unstable software packages. The strategy reduced the odds of having technical problems and consequently lessened the adverse impacts of upgrade.

7.2.1.3 Business Needs

With IT serving business users, it is understandable that demand from the business community can be an important motivator. Almost all internal requirements are stamped from business needs to ensure a smooth and continuous business operations. Even though not all business needs trigger an upgrade, an imminent one can. In this study, business needs prompted

SAPSD to look into the new version of SAP and they found some potentially useful functionality that became one of the motivations for the SAP 4.6C upgrade. Likewise, the need to integrate business systems after CP's acquisition of NAC in 2000 prompted the division-wide upgrade to Windows 2000.

7.2.2 External Dependency on Software Vendor

When organizations adopt packaged software, they become dependent on the software vendor to provide them with software functionality and technical support. In the old days, when system maintenance was mainly studied on in-house developed systems, three groups were involved in system maintenance: application systems, IS staff, and users (Swanson and Beath 1989). However, with packaged software, the number of groups expands to include the vendor that the organization relies upon for IT needs.

An organization's dependency on software vendors for some of its most important IT resources gives vendors influence over an organization's upgrade decision. According to Pfeffer and Salancik (1978), "It is the fact of the organization's dependence on the environment that makes the external constraint and control of organizational behavior both possible and almost inevitable" (p. 43). The extent to which the organization complies with external demands depends on three elements: (1) whether the resource is important, (2) whether the external company has "discretion over the resource allocation and use," and (3) whether there are other alternatives for the resource (Pfeffer and Salancik 1978).

Both the Windows operating system and SAP system were extremely important to CP. One was the backbone for the entire business operation; the other was the platform that ran most of the business applications. Because both software packages were critical to CP's continued

operation and survival, the threat of losing support and not receiving technical assistance was inconceivable to CP. As a result, the vendor's sunset date was one of the most important motivating influences in an upgrade decision. Along with company policy that required all packaged software to remain on technical support, the sunset date become the ultimate deadline for IT to upgrade, as shown in the SAP 4.6C case.

7.2.2.1 Functionality of Software

One important element that organizations look for in deciding whether to upgrade is the functionality in new version that can benefit them. In fact, the relative advantage of new technology has long been recognized as an influence on individual adoption in the innovation diffusion literature (Rogers 1983).

In both cases, functionality was a strong motivating influence but not the trigger for the decision to upgrade to SAP 4.6C or Windows 2000. The SAPSD saw new functionality that they could adopt to meet business needs, and the Infrastructure Group recognized tremendous benefits in adopting Active Directory. Although a significant influence, functionality did not "evoke" (Mintzberg, Raisinghani and Theoret 1976) the upgrade in either case. In both cases, a working system already existed. As long as the current version was functional and there was no immediate urgency to perform the upgrade, under the situation of scarce resources, CP's IT usually waited to upgrade even though the new functionality was cited as having many benefits. Since both of the cases studied involved scarce resources, it is unclear if the outcome would have been different if resources were abundant. The next section will look at internal resources.

7.3 Internal Resources

Internal resources are one of the most significant influences on upgrade decisions. According to Mohr (1982), "Motivations may determine behaviors, for example, but only if the relevant resources are adequate." Although the availability of resources by themselves may not trigger or promote the decision to upgrade, the lack of resources often means that upgrade initiatives remain on the ground. Resource availability can have a huge influence on an organization's ability to carry out a project. This conclusion is supported by evidence from the two cases. Although the Infrastructure group saw the benefit of Active Directory, it did not upgrade sooner due to lack of resources. Likewise, SAPSD did not upgrade when their "18 months" policy hit due to lack of internal resources.

Although the study agrees with Mohr (1982) that "resources interact with motivation in determining outcomes," it does not share Mohr's (1982) view that resources dominate the decision process. According to Mohr (1982), "If the resources for carrying out the action are insufficient, then there is not likely to be much difference between the behavior of subjects who are highly motivated and those who are not." In the study, although lack of resources can indeed defer upgrade decisions when such deference does not constitute an immediate threat to business operations, the decision is different when there is an urgent need to upgrade. The next section looks at how interaction between motivating influences and internal resources affects the upgrade decision.

7.4 Interaction of Motivating Influences and Internal Resources

In both cases, more than one stimulus was observed along the decision process to influence the decision to upgrade. However, not all of them exerted equal amounts of force. According to Mintzberg et al. (1976), each stimulus has amplitude. A decision is "evoked" when "cumulative amplitude of stimuli" reaches the "action threshold." In the two cases, amplitudes and action thresholds were not explicitly assessed as part of the decision process. The threshold level was a subjective reference point made by managers; no formal cost/benefit analyses were conducted at CP. Nonetheless, it can be inferred from managers' comments that the upgrade decisions for both projects were affected by multiple influences, as shown in the model. According to Bannister and Remenyi (2000), the "gut feel" approach is in fact a quite common phenomenon in organization decision making. Similarly, Mintzberg et al. (1976) found that the favored mode chosen by individuals when evaluating alternative solutions was to make "a choice in his own mind with procedures that he does not, perhaps cannot, explain."

From the two cases, the threshold can be inferred as a point at which managers perceived the need to upgrade had become a necessity and was justifiable for the scarce resources. When the decision was made to upgrade SAP, the software was in "crisis" (Mintzberg et al. 1976) of losing support and had to be tended to immediately. In the Windows case, the Infrastructure Group was finally able to justify the division-wide upgrade to Windows 2000 when the need arose to replace hardware for NAC. The combination of stimuli (in a subjective sense) provided the needed justification to conduct the upgrade.

Even though no project can be executed under a complete void of resources, resource scarcity does not mean that no project can be executed at all. From the two cases, scarcity represents a situation in which IT is unable to attend to all desired projects. Under this

circumstance, only projects deemed most critical by managers will be prioritized and scheduled to receive the needed resources.

In the study, motivating influences and internal resources interacted to form different decision outcomes depending on the level of motivation intensity and internal resource sufficiency. In the situation where resources are scarce, the availability of resources dominates the outcome of interaction, particularly when the need to upgrade can be deferred. On the contrary, the need to upgrade dominates the decision outcome when the stimulus becomes an imminent need or has reached the threshold (Mintzberg et al. 1976).

While the above findings pertain to decision making under resource scarcity, it is believed that under the circumstances in which resources are sufficient, any legitimate motivation would evoke the upgrade decision. The claim is supported by previous upgrades patterns. In the past, when the SAP department had fewer competing projects, upgrades were performed according to 18 months policy.

Upgrades inevitably produce impacts on different stakeholders. The next section provides discussion on types of coping strategies and how adoption of those strategies moderates the final impacts of upgrades.

7.5 Coping Strategies

A packaged software upgrade, like any IT project, is risky and can create unintended consequences. With a version already in use and running day-to-day operations, CP's biggest concern was the potential adverse effect from undertaking the upgrade project. Upgrade coping strategies were formulated on the premise of minimizing adverse impacts from the upgrades. Although some degree of productivity loss was expected in each upgrade, the organization

wanted to minimize disruptions to business operations. To achieve that, at CP, an upgrade was usually conducted with three goals in mind: making sure new system migrated without problems, deploying the system with minimal down time to users, and making sure that users were able to perform daily operations after the upgrade.

Unlike decision making, in which the emphasis of the process is usually put on the potential benefits of migration, the focus of implementation is often placed on minimizing the problems of upgrade. The assumption is that positive effect will materialize if the project is implemented properly.

From the study, two types of coping strategies were observed: a planned strategy that proactively addressed anticipated issues in an upgrade, and corrective action that reacted to unanticipated issues that arose during upgrade. Most strategies observed in both studies fall under the category of planned strategy. Both strategies will be discussed in the following sections.

7.5.1 Planned Strategies

According to Schmidt et al. (2001), studies on project management and IS implementation deal with the subject of project risk. Software project risk is defined as "a set of factors or conditions that can pose a serious threat to the successful completion of a software project" (Wallace, Keil and Rai 2004). According to Boehm (1991), software project disasters "would have been avoided or strongly reduced if there had been an explicit early concern with identifying and resolving their high-risk elements…before they become either threats to successful software operation or major sources of software rework" (p. 32).

Although CP did not identify the strategy that they used as risk management, their approach was similar to the underlying principle of risk management, which is to address threatening conditions before they become serious problems. Even though areas of risk were not explicitly mentioned during the study, they can be inferred from the strategy prescribed.

IT at CP carefully devised strategies to counter any issue that they could foresee. The focus of planned strategies is to preempt any issue that could arise from the upgrade, consequently reducing the potential adverse impact of the project. In this study, coping strategies were found to have moderating effects on the outcomes of software upgrade decisions. The adoption of a specific strategy can either increase the positive impacts of upgrade or decrease the negative consequences. The next section will look at some of the major strategies that IT employed and how those strategies moderated the final outcome of upgrade.

7.5.1.1 Coping Strategies for Issues in Packaged Software

In an upgrade, one of the potential areas of risk is the packaged software itself. Besides the renowned fact that software is released with bugs, one of the most common issues in technology change is compatibility (Lederer and Mendelow 1990). As a software vendor moves forward with technology, new releases may no longer be compatible with existing applications of an organization. Furthermore, an upgrade is especially cumbersome when customization has been introduced to the existing software. Changes to internal code are almost guaranteed to be overwritten when a new version of software is loaded. From the study, two key strategies were used to cope with packaged software upgrade issues. Each will be discussed in the following paragraphs.

Testing

According to Felix (1984), testing is so important that it deserves to have a stage in project management. Testing not only allows IT to identify bugs present in the software, but also permits them to discover changes in existing functionality. IT also tests for compatibility of new versions with existing applications. Ensuring the quality of the upgraded system is especially important for critical business applications like SAP, where the platform is virtually the business environment, and any issue can only affect business operation. With control over development lying in the hands of the vendor, not all changes are communicated to organizations, so testing becomes a crucial tool that IT uses to identify issues that needed remedies before the new version is put into production.

Because both systems served very different purposes, the emphasis on testing differed according to the types of software. Compatibility issues were critical in Windows 2000 because it was an operating system upgrade, whereas correct functionality was important in SAP where business operations were run on the platform.

In both cases, extensive testing was conducted. Windows 2000 was tested by corporate IT against all commonly used products before it became a corporate standard. Later, when CP decided to adopt it, six more months of testing was conducted on local software. On the other hand, SAP had over 2,000 test cases that IT used to determine the intactness of commonly used functionality. Testing was also very important for SAPSD due to customization that they did in the software.

Testing allows both IT departments to identify and address various problems to reduce the chances of having troubled system. In general, the positive impacts of testing are quite difficult to assess due to the opaqueness of the effects produced. Because a successfully tested

system is least likely to create problems, the potential consequences of not testing the system are never be known, except to say that system works without problems.

No Customization

While standard software applications like Word and Excel seldom require customization, tailoring SAP to company needs is almost unavoidable. Customization brings higher costs of testing and re-development, which affects the final cost of implementation. Hence, the SAPSD department had a policy that greatly discouraged customization unless it was absolutely necessary.

Of the two strategies given, “no customization” was a departmental policy put in place to prevent an issue from happening, whereas “testing” allowed IT to detect issues that could not be prevented and to find the appropriate solution. While attempting to implement an error-free packaged software is very challenging and almost unattainable, coping strategies described in this section can indeed reduce the incidence of troubled systems.

7.5.1.2 Implementation

Besides making sure packaged software functioned as desired, IT also paid considerable attention to the software deployment process. An inadequately planned deployment process is a potential breeding ground for upgrade problems. In their study to identify software project risks, Schmidt et al. (2001) found that “lack of effective development process/methodology” leads to quality problems. Deployment involves many steps, from data migration to integration with other systems; IT’s strategy is to plan ahead on every detail of the deployment process. IT from both departments at CP was aware of the consequence that a misstep could bring to the division.

SAPSD conducted multiple mock upgrades to make sure migration proceeded as planned, and the Infrastructure Group used checklists to make sure the user's software was properly loaded. In addition, Desk Side Support personnel also kept users' old PCs for a week after migration as a precaution against data loss.

Although the same implementation philosophy was adopted in both cases, due to technology differences, SAPSD and ISD differed in their ways of deployment. To reduce system downtime, SAP conducted the deployment during a weekend when production was at a lower level. The Infrastructure Support Department adopted "build and swap" to minimize PC downtime and to reduce the time IT personnel spent at users' desks.

Overall, the strategy was to have a well-planned deployment process to address any issue that IT could conceive before actual implementation took place. The extent of planning went as far as anticipating vendor's failed support. According to Benamati and Lederer (2001), relying on vendors for technical support is the fourth most used coping strategy when technology changes. However, Schmidt et al. (2001) also pointed out that software projects can be risky when vendors do not deliver. In both cases, IT personnel relied on vendors for technical issues they could not resolve. In the SAP case, past experience was that the vendor was slow to respond to their problems. So SAPSD coped by developing their own strategies just in case vendor's help did not arrive in time for deployment.

7.5.1.3 Learning Assistance

According to Markus et al. (2000), "It is important not just how well the ERP system itself performs (e.g. accuracy, reliability and response time), but how well people in the

organization know how to use … the system." Although the study focused on ERP applications, the statement is true for all systems.

At CP, to ensure continuous business operation, IT not only paid attention to system quality but also focused on users' ability to perform daily tasks after implementation. Both SAPSD and ISD provided learning assistance to IT and users as part of their upgrade strategies.

From both cases, slightly different assistance was given based on the changes introduced in the upgrade. Although the Windows 2000 upgrade was considered a huge change in underlying technology, it was not perceived by IT to be a big change in the front end. As a result, only reference (cheat) cards were given to guide users through the changes presented in the software. On the contrary, mandatory in-class training was used in the SAP case because the new version was seen by both IT and users as a brand new application.

According to Benamati and Lederer (2001), education and training is one of the coping strategies used to deal with technology changes. The importance of training has been stressed in many studies (Lientz and Swanson 1981; Bostrom, Olfman and Sein 1990; Robey, Ross and Boudreau 2002; Somers and Nelson 2004; Nah et al. 2001). In their study, Nelson and Cheney (1987) found a relationship between computer-related training that users received and their ability to use the computer resource.

In fact, SAPSD was so concerned with user training that when IT ran out of resources, business units trained their own employees after IT "trained the trainers." In the SAP case, training was mandated for all affected employees. In addition to training, both SAPSD and ISD had technical support structure in place at all times to assist users. Learning assistance was more intense in the SAP upgrade than in the Windows 2000 case. One explanation is that SAP was a

mission critical system and change was indeed larger in SAP. IT and business leaders wanted to make sure that users were able to perform business operations after the implementation.

In the SAP case, additional help was given to users with SAP personnel “walking the floor” to provide technical assistance the day after the upgrade. Business managers also made rounds to give users moral support and encouragement. All of the learning assistance was provided to improve users’ ability to use the upgraded version and reduce productivity loss.

7.5.1.4 Users’ Strategy

Although users received learning assistance for each upgrade, users in both cases found hands-on learning to be most effective, citing “trial and error” as their preferred strategy when it came to coping with software upgrade changes. One plausible explanation is after many years of working with computers, users are less anxious (Thatcher and Perrewe` 2002) with new technology, more confident with computer use, and no longer afraid to explore. However, users’ explorations did not replace formal training. Rather, trial-and-error learning complemented the knowledge gained from attending formal training and reinforced their capability to perform tasks after the upgrade.

In both cases, users portrayed themselves as comfortable with technology. Almost all users of Windows 2000 said they were able to transition without much problem, as did users of SAP. However, while no users claimed to be having trouble, trainers and some users observed that change was actually easier on those with fewer years of experience. The more years a user works with a specific technology, the more technology-specific skills become stamped on one’s memory, making a shift from the usual ways of doing things more difficult for those users.

7.5.2 Coping Strategy for Unanticipated Events

As much as IT planned for a smooth transition, there were a few unexpected events that they had to cope with. Some were quickly resolved, e.g., missing information on reports as a result of internal code changes by the SAP vendor. Others drew more attention and required a reexamination of strategy used, e.g., the Active Directory crash. In that incident, IT revised policy to restrict unnecessary access to Active Directory, and set up tracking mechanism that later became part of a permanent prevention strategy.

When confronted with unplanned issues, IT's solution was to react as quickly as possible to the problem and provide corrective action to remedy the situation. While not all, many of the lessons learned and solutions developed from past issues were incorporated into planned strategies to improve performance on subsequent upgrades. SAP personnel explained that they continued to learn and improve their upgrade process based on past experience.

7.6 Impacts

Every upgrade decision inevitably incurs costs and, hopefully, derives benefits. Although considerable value can be obtained when useful software features are adopted, problems during upgrades can be costly too. No hard figures were collected on the impacts of upgrade in either project due to the difficulty of assessing the impact. Nevertheless, findings from the two cases revealed both positive and negative impacts of upgrades. The discussion of upgrade impacts is organized according to implementation process, changes in packaged software, and circumstantial impacts.

7.6.1 Impacts Originated from Implementation Process

Cost of implementation has often been ignored in the IT evaluation literature. According to Melville and Kraemer (2004), “Based on our analysis of the IT business value literature, there is no convention regarding the incorporation of costs of system development and implementation.” Most of the costs of upgrade are associated with implementation processes.

In the two cases studied, IT personnel saw upgrades as huge projects that consumed a tremendous amount of their resources. SAP took approximately six months, and Windows 2000 took more than a year to complete. Their experience is in line with the experience of other SAP customers in their industry. According to Kremers and van Dissel (2000), “Many migrated customers perceived migrations as difficult, costly, and above all as time-consuming processes.” Although upgrades have consistently been thought as smaller scale projects compared to initial implementations, they may still consume large amounts of resources. One analyst described the difficulty of an upgrade: “It's still nothing that you take out of a box and go right to work. It’s gone from a migraine to a headache” (Weston 1997). One reason that upgrades are difficult to perform involves the changes that new versions introduce. Often, new versions are said to be “evolving in terms of technology and functionality” (Kumar and Hillegersbery 2000) and to be “profoundly different” (Kremers and van Dissel 2000). The bigger the change, the more challenge IT faces.

Both upgrades studied were huge challenges to IT. SAP was a multi-version upgrade whereas Windows 2000 Active Directory was a paradigm shift for IT personnel. Although the scope of implementation was influenced by the amount of changes in the new versions, the impact of implementation, both length of period and the final outcome, were greatly influenced by the coping strategy that CP employed. A strategy that calls for a careful implementation takes

longer but produces quality output and reduces adverse impact compared to a hasty implementation that may not lead to a success in implementation.

Because CP's policy had low tolerance for IT project failure, the SAP Support and Development Department took a very conservative and cautious strategy that required detailed testing and several trial runs to ensure the installation worked without problems. The Infrastructure Support Department, however, spent at least six months to test the software before beginning to migrate users to the new operating system. While the strategy required a longer period of implementation, the director of SAPSD department attributed the positive impacts to the coping strategy that they adopted.

7.6.2 Impacts of Product

Packaged software upgrades often introduce two common types of change: functionality and user interface. Functionality changes include the addition, removal, or modification of existing software capability. User interface change is usually done to improve usability of the software and affects the "look and feel" of the front end. In user interface changes, functionality can remain the same but the front end used to access the functionality can change dramatically.

From the study, different perceptions of software impacts were reported, whether it was between different stakeholders experiencing same technology upgrade or within one stakeholders group experiencing different technology upgrades. This comes as no surprise because "different aspect of software is important to different users" (Brinton, Akhilesh and Gorr 2002). In general, the impacts that stakeholders perceived were closely linked to the roles that they played in the upgrade, either as an implementer, user, or both. As an implementer, IT personnel's experience was often connected to changes in the back end. By contrast, users' experience was usually

associated with improvements in the front end. Moreover, a stakeholder can experience different impacts based on the types of technology being upgraded. Three different technologies were upgraded in the two cases: SAP 4.6C, Windows 2000 and Office 2000, each catering to different business needs.

To understand a software product's impacts on stakeholders, the discussion focuses on two of the biggest impacts in the study: functionality improvements that the Windows 2000 upgrade brought to IT personnel, and user interface changes that impacted SAP users.

Windows 2000 was a huge technology improvement from IT perspective. The operating systems was much more stable, and features like net meeting and self-healing capability cut down the support time needed for IT personnel. Most importantly, Active Directory completely overhauled back end software management, automated software distribution, and eliminated the need to manually install upgrades or service packs. Although the impact that IT personnel perceived was tremendous, most users did not notice much improvement from the upgrade.

The discrepancy of perceptions may seem strange at first, but careful examination reveals that in order for functionality improvements to exert effect, they have to be either adopted, if the feature is new, or affect existing functionalities that are used. Thus, an upgrade can only impact stakeholders if the change is relevant to their software use, and the impact is visible. In the Windows 2000 upgrade, most of the features that brought tremendous improvement to IT were not relevant to users and were not perceived.

In both cases, users did not perceive much functionality improvement in any of the three technologies. In SAP, functionality was not introduced as part of the implementation strategy in order to minimize confusion for business users. When upgrading to Windows 2000, some of the users previously on Windows 95 noticed the system did not crash once a day. However, many

users of Windows NT 4.0 did not notice this difference because NT 4.0 was a pretty stable system itself. Finally, for Office applications, most users only used standard features that remained intact following the upgrade. Of all three technologies upgraded, users did not explore new functionality, although functionality was one of the key motivations for making the upgrade decision.

The key focus after the upgrade was users' ability to perform their previous tasks. Whether new features added value or not was a secondary concern to users compared to regaining their ability to perform routine tasks. Any change presented in the software that deviated from the usual way of processing a task imposed learning curve. From both cases, the biggest impact that business users perceived from the upgrade was not the benefits received from improved features, but the time and effort invested to return to their normal work processes.

Besides functionality, the second component that can potentially affect users greatly is interface changes. In the SAP 4.6C upgrade, the user interface changed from menu-based to icon-based, and the front end looked dramatically different after the upgrade. Although the changes were seen as improvements by many business managers interviewed, most users merely saw them as "changes" and not as improvements.

User interface changes can create very significant impacts on users because the software interface is where users interact with the system. In the world of the electronic workspace, the user interface is essentially the work environment for business users. When the interface changes, the work environment changes, and the impact can be substantial.

A change in the way a task is processed can easily disrupt the workflow. For many, keystrokes are ingrained in their daily routines. When a button is moved from its usual place, users have to find where to click instead of just clicking in a familiar place. In fact, a temporary

drop of productivity was expected following the SAP upgrade, and was negotiated in advance as part of an upgrade agreement between SAPSD and business top management.

On all three technologies, user interfaces changed to a different extent. A few users noticed the login screen changed for Windows 2000, and the user interface was slightly different in Office applications like Word and Excel. Overall, it took users about two weeks to adjust to Windows 2000 and three months to adjust to SAP 4.6C. Generally, the length of learning period needed is dependent on the degree of change that users experienced.

7.6.3 Circumstantial Impacts

Circumstantial impacts refer to impacts that are not directly related to the software itself or to the implementation process, but which still result from upgrade decisions. For example, the interconnectivity of packaged software with peripheral systems can cause additional impacts if incompatibility issues are discovered during testing. In order to move ahead with Windows 2000, ISD personnel had to replace all print drivers to be Windows 2000 compatible, and the logistic department had to scramble to find a solution. During the Windows 2000 upgrade, ISD also experienced the worst scenario when Active Directory crashed. Although it was considered a good learning experience that later improved the process of managing Active Directory, the financial impact was estimated to be in the millions of dollars.

CHAPTER 8

CONCLUSIONS

The motivation for this study was to investigate the decisions, coping strategies, and impacts of packaged software upgrades from the perspectives of stakeholders. Using the case study method, two cases of packaged software upgrade on two technologies were examined: SAP and Windows operating systems, within a large division in a *Fortune* 500 company.

From the study, decisions to upgrade were found to be governed by both the availability of internal resources and motivating influences originating from internal and external environments. Because an existing version was already in place, performing daily operations, a heavy emphasis was put on ensuring a successful implementation that brought the least disruption to the user community. Strategies ranging from preventive policy restricting the use of unstable software, to extensive testing for identifying and resolving issues ahead of actual deployment, were employed. In general, the impacts of the upgrade were substantial. However, not everyone received the same upgrade impact. Different results were observed based on types of technology that stakeholders used and the degree of change that affected the functionality that they used. The two case studies were presented in detail in Sections CHAPTER 5 and CHAPTER 6. In Section CHAPTER 7, a theoretical model based upon the results of a cross-case data analysis was constructed. The model is comprised of six components: decision, motivating forces, contingency forces, planned strategies, corrective actions, and impacts. The decision to upgrade is influenced by motivating forces and contingency forces, and leads to positive and negative impacts that are moderated by planned strategies. Corrective actions respond to unanticipated impacts.

8.1 Implications for Research

Four implications for research can be claimed. First, the research is one of very few studies that focused on packaged software upgrades. Through two case studies, this research provides exploratory results on decisions, impacts, and coping strategies to a field that has not received much attention. Besides allowing readers to learn more about each of those three areas in packaged software upgrade, investigating all three areas in one study also provides a synergy that allows the interrelationships among those areas to emerge and be understood. Given the growing importance of packaged software, and the inevitability of upgrades, it is increasingly necessary to understand the decision processes, coping strategies, and impacts of packaged software upgrades.

In the study, both departments achieved most of their intended outcomes. SAPSD adopted an incremental approach (Robey et al. 2002) to their upgrade and did not seek functionality improvements. By contrast, ISD received immense back end and software improvements that benefited IT personnel because a new technology, Active Directory, was adopted. Moreover, planned coping strategies were found to mitigate the risk of negative impacts, consistent with the advice of the software project and risk management literature.

Second, although a software upgrade is a type of IS project, it has some salient characteristics that separate it from traditional IS projects, which either focus on system development or initial adoption of a commercial system. In a packaged software upgrade, a version of the system has already been implemented and accepted by users. Adding to that was the fact that vendor has control over the development of packaged software. In upgrades, organizations become dependent on software vendors for support and modifications. This dependency makes organizations vulnerable to vendor's actions, whether it is setting sunset dates

or having slow response to organizational needs. Sunset dates have an important influence on the CP upgrade policy and slow vendor response can affect the implementation timeline, according to the study. The study observed an increase in vendor's influence and control on the organization's IT policy.

Although both IT departments adopted strategies similar to risk management, some of the strategies used in traditional IT implementation (e.g., user participation) were not as important to upgrade as in system development. In the SAP case, user participation was limited to pilot testing of the new version. In contrast, the upgrade strategy at CP placed an important emphasis on reducing adverse impact of upgrade.

As for impacts, because users already had a functional system, their biggest anticipation in an upgrade was seldom functional improvements but rather their ability to continue performing routine tasks. Unlike initial adoption in which one of the impacts anticipated would be improvements brought by the new system, in an upgrade, functionality improvement was usually not the key impact. Although the Windows 2000 upgrade saw huge benefits from IT's perspective, most previous upgrades to operating systems were conducted for bug correction, not the significant functional improvements in Windows 2000. In an upgrade, the biggest impact for IT is often their time and effort whereas the biggest impact for users is usually the learning curve. Neither of these impacts has received much mentioned in IS evaluation literature.

Third, this research investigated packaged software upgrade from different stakeholders' perspectives. By probing multiple stakeholder groups, the study provides a better understanding of the outcomes of software upgrades. While not all stakeholder groups played equal parts in all three areas probed, their perspectives are nevertheless important to help readers understand the phenomena investigated, and also to provide a more diverse view, not just perceptions from one

party. Overall, there is a clear role difference between the two groups of stakeholders at CP: IT and users. IT managed the upgrade process and bore the burden of implementation, whereas users' roles are focused on interacting with the systems. Thus, IT's experience with the upgrade was often associated with back end improvements whereas users' experiences were affected by front end changes. Although stakeholders' experiences differ, they are interdependent. In the study, IT's efforts minimized the potential for the upgrade to produce adverse impacts on the users, allowing users to perform their tasks without serious disruptions. Thus, the impacts of software upgrades are experienced differently by different stakeholders, reinforcing the importance of the stakeholder approach to research.

A fourth contribution of this study is the richness of detail that the two cases provide in uncovering various upgrade issues in different technologies. Although data are context sensitive to one large organization, the cases nevertheless provide a starting point for research into packaged software upgrades, which has not received much attention.

Though flaws of this research are acknowledged in the following limitations section, the study is one of the first that looked into packaged software upgrades in depth. Like any exploratory study, findings are preliminary and more efforts are needed to study the phenomena further.

8.2 Limitations

There are several limitations to this research that should be noted. First, like any case study, generalizing from the research findings should be cautious. Even though two cases were used to induce the model for this study, both cases were drawn from the same large division of a *Fortune* 500 company. There is substantial possibility that some practices observed in the

research are context specific to SE Co. or specific to large corporations. Also, the findings are limited to the two different technologies studied and should not be generalized to all packaged software.

The second limitation of this study is inherent in the data collection method that I used. For a researcher, adoption of the interview method carries the risks of not receiving accurate answers due to the subject's memory lapse or the subject's reluctance to share the truth on certain areas. With interviews as the primary source of data generation, I was aware of the "reactive" (Stone 1991) nature of the technique. During the interviews, there were a few occasions when subjects questioned my status as an employee because I wore a contractor's access pass. I noticed subjects responded more freely when they felt they were talking to an outsider than an employee of the company. Because I was required to wear the pass at all times, I could not remove it during interviews. However, upon discovering the deviation of attitude, when I conducted the interview, in addition to my usual introduction that I was a doctoral student from Georgia State, I would explain the pass was just a means that allowed me temporary access and I was not affiliated with SE Co. in any way.

Another limitation of using the interview method was my reliance on subjects' memories and accounts of events and my inability to observe events unfold. The concern was more relevant in the SAP case. Because the project was completed a year and a half before the study, I depended on subjects' recounts of events for the majority of my data collection. As Kirsch (1997) pointed out, retrospective accounts can "result in some distortion of the facts." Although most IT personnel recalled the events vividly because the project was a huge undertaking, a few users got confused with the facts in upgrade and some functionality improvements projects that were conducted earlier. To address these problems, I triangulated the accounts of the subjects

with their counterparts, and also compared accounts across stakeholder groups to make sure subjects were indeed referring to SAP 4.6C upgrade when they responded. In situations where there were suspicions of inaccurate information, I conducted follow up interviews. I also solicited more users for interviews and expended more effort to look for discrepancies among data in the SAP case. As Leornard-Barton (1990) suggested, the researcher using retrospective reports to investigate historical events has to "work harder to be a critical audience" (p. 257).

8.3 Implications for Practice

Any organization that uses packaged software can benefit from the findings of this study. First, the study provided various strategies that IT could adopt to cope with upgrade issues. According to the study, strategies should focus on three areas that, if neglected, could create problems: the packaged software itself, the implementation process, and the users' post-implementation use. For each of these areas, potential issues that could arise and strategies used to cope with those issues were given to provide a better understanding of how problems could be averted to ensure a more favorable outcome. Because the case also looked at impacts of upgrade, the study provided valuable information on how the strategies affected the final outcomes of the upgrade.

Second, this research provided an example of how upgrades were carried out in a huge division of a *Fortune* 500 company. The study recorded two cases of upgrade explicating stakeholders' experience with upgrades of two different technologies. Some of the practices reported in the study can serve as guidance on how to conduct upgrades, and lessons from the cases can be warnings to others not to repeat the same mistakes. The information gathered in this study could be referenced by organization seeking to improve their upgrade process.

Nevertheless, the limitations section above cautions readers of this research to understand that the results are context sensitive. Because one of the advantages of a case study resides in the bountiful details that it provides, IT personnel intending to use the information from the two cases should review the background information provided and determine if the results are applicable to their environment.

8.4 Future Research Directions

With the topic of packaged software upgrade just beginning to attract research attention, this exploratory study offers some insights into how future research might proceed. The current research asked three questions targeting decisions, impacts, and coping strategies of software upgrade. Even though preliminary findings were presented in each area and a model was induced from the overall study, the research design of two cases restricts its applicability to the general upgrade phenomena. More research is required to take the findings of this study and confirm the results observed. To begin, additional research using case study and survey methodologies should be performed in different settings to verify and extend the findings in this study.

New research could target each of the three research questions presented in the study to achieve a deeper understanding for each area. For the upgrade decision, one potential research study would be to investigate if, internal resources were ample, would the interaction between motivating influence and internal resources remain the same. Moreover, under the hypothetical circumstance, which motivating influence would dominate the upgrade decision? Additional research can also be conducted to uncover a common set of strategies that would be useful for packaged software upgrades. The proposed research can supplement the software risk and project management literature that usually focuses on in-house software development and initial

adoption of packaged software. Aside from those two areas, future study could also focus on understanding the impacts of packaged software upgrade beyond perceptual data and provide a more concrete measure for the actual impacts that upgrades bring.

Packaged software upgrades will become a more important topic as practitioners and researchers realize the extent of impact it exerts on organization. At the moment, the impact was felt by the time required to perform the upgrade. With almost all companies adopting one or more types of packaged software, each requiring a version upgrade that could easily span six to nine months (according to the current study), the toll taken on IT management and resources will be substantial. Research is needed to understand and better manage the situation that will continue as long as packaged software is adopted.

REFERENCES

Alavi, M. and E. A. Joachimsthaler (1992). "Revisiting DSS Implementation Research: A Meta Analysis of the Literature and Suggestions for Researchers." MIS Quarterly **16**(1): 95-116.

Anonymous (2000, January 25). "IDC Puts the Worldwide Packaged Software Market at $154 billion in 1999" from http://www.idc.com/software/press/PR/sw012500PR.stm

Baig, E. C. (2001, May 16). Microsoft's Office XP: More than `eXtra Parts'. USA Today. Arlington, VA.

Bannister, F. and D. Remenyi (2000). "Acts of faith: instinct, value and IT investment decisions." Journal of Information Technology **15**: 231-241.

Barua, A., C. H. Kriebel, et al. (1995). "Information Technologies and Business Value: An Analytic and Empirical Investigation." Information Systems Research **6**(1): 2-23.

Belcher, L. W. and H. J. Watson (1993). "Assessing the Value of Conoco's EIS." MIS Quarterly **17**(3): 239-253.

Benamati, J. and A. L. Lederer (2001). "Coping with rapid changes in IT." Communications of the ACM **44**(8): 83-88.

Benamati, J. and A. L. Lederer (2001). "Rapid Information Technology Change, Coping Mechanisms, and the Emerging Technologies Group." Journal of Management Information Systems **17**(4): 183-202.

Blankenship, L. V. and R. E. Miles (1968). "Organizational Structure and Managerial Decision Behavior." Adminstrative Science Quarterly **13**(1): 106-120.

Boehm, B. W. (1991). "Software Risk Management: Principles and Practices." IEEE Software **8**(1): 32-41.

Bostrom, R. P., L. Olfman, et al. (1990). "The Importance of Learning Style in End-User Training." MIS Quarterly **14**(1): 100-119.

Bray, H. and G. Staff (2001, June 9). Glitch Halts NYSE Trading Temporary Shutdown Hurts Retail Trades, Affects Other Markets. Boston Globe. Boston**:** C1.

Brinton, B. A., B. Akhilesh, et al. (2002). "An estimation of the decision models of senior IS managers when evaluating the external quality of organizational software." The Journal of Systems and Software **61**: 59-75.

Burrell, G. and G. Morgan (1979). Sociological Paradigms and Organizational Analysis. London, Heinemann Educational Books Ltd.

Choudhury, V. and J. L. Sampler (1997). "Information Specificity and Environmental Scanning: An Economic Perspective." MIS Quarterly **21**(1): 25-53.

Creswell, J. W. (1997). Qualitative Inquiry and Research Design. Thousand Oaks, CA, Sage Publications, Inc.**:** 47-72.

Denzin, N. K. and Y. S. Lincoln, Eds. (1998). Entering the Field of Qualitative Research. The Landscape of Qualitative Research Theories and Issues. Thousand Oaks, Sage Publications.

Eisenhardt, K. M. (1989). "Building Theories from Case Study Research." Academy of Management Review **14**(4): 532-550.

Felix, R. G. (1984). "Project Management Considerations for Distributed Processing Applications." MIS Quarterly **8**(3): 161-170.

Finlay, P. N. and A. C. Mitchell (1994). "Perceptions of the Benefits From the Introduction of CASE: An Empirical Study." MIS Quarterly **18**(4): 353-370.

Freeman, R. E. (1984). Strategic Management. Boston, Pitman.

Glaser, B. G. and A. L. Strauss (1967). The Discovery of Grounded Theory. Chicago, Aldine Publishing Company.

Goodhue, D. L., M. D. Wybo, et al. (1992). "The Impact of Data Integration on the Costs and Benefits of Information Systems." MIS Quarterly **16**(3): 293-311.

Guba, E. G. and Y. S. Lincoln (1998). Competing Paradigms in Qualitative Research. In N. K. Denzin and Y. S. Lincoln, The Landscape of Qualitative Research Theories and Issues. Thousand Oaks, Sage Publications.

Harrison, D. A., P. P. J. Mykytyn, et al. (1997). "Executive Decisions about Adoption of Information Technology in Small Business: Theory and Empirical Tests." Information Systems Research **8**(2): 171-195.

Hinton, M. and R. Kaye (1996). "Investing in Information Technology: A lottery?" Management Accounting **74**(10): 52-55.

Hitt, L. M. and E. Brynjolfson (1996). "Productivity, Business Profitability, and Consumer Surplus: Three Different Measures of Information Technology Value." MIS Quarterly **20(**2**)**: 121-142.

Hoxmeier, J. A. (2000). "Software Preannouncements and Their Impact on Customers' Perceptions and Vendor Reputation." Journal of Management Information Systems **17**(1): 115-139.

Kim, C. and S. Westin (1988). "Software Maintainability: Perceptions of EDP Professionals." MIS Quarterly **12**(2): 167-185.

Kirsch, L. J. (1997). "Portfolios of Control Modes and IS Project Management." Information Systems Research **8**(3): 215-239.

Klein, H. K. and M. D. Myers (1999). "A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems." MIS Quarterly **23**(1): 67-93.

Kremers, M. and H. van Dissel (2000). "ERP system migrations." Communications of the ACM **43**(4): 52-56.

Kumar, K. and J. V. Hillegersbery (2000). "ERP Experience and Evolution." Communications of the ACM **43**(4): 22-26.

Leana, C. R. and B. Barry (2000). "Stability and Change as Simultaneous Experiences in Organizational Life." Academy of Management Review **25**(4): 753-759.

Lederer, A. L. and A. L. Mendelow (1990). "The Impact of the Environment on the Management of Information Systems." Information Systems Research **1**(2): 205-222.

Lee, H. G., T. Clark, et al. (1999). "Research Report. Can EDI Benefit Adopters?" Information Systems Research **10**(2): 186-195.

Leonard-Barton, D. (1990). "A Dual Methodology For Case Studies: Synergistic Use Of A Longitudinal Single Site With Replicated Multiple Sites." Organization Science **1**(3): 248-266.

Lientz, B. P. and E. B. Swanson (1981). "Problems in Application Software Maintenance." Communication of the ACM **24**(11): 763-769.

Littman, D. C., J. Pinto, et al. (1987). "Mental Models and Software Maintenance." Journal of Systems and Software **7**(4): 341-356.

Lucas, H. C., E. J. Walton, et al. (1988). "Implementing Packaged Software." MIS Quarterly **12**(4): 537-549.

Maier, J. L., K. Rainer Jr., et al. (1997). "Environmental Scanning for Information Technology: An Empirical Investigation." Journal of Management Information Systems **14**(2): 177-200.

Manes, S. (2001). The ungrade path: Uphill and rocky. PC World. **19:** 234-235.

Markus, M. L., S. Axline, et al. (2000). "Learning from adopters' experience with ERP: problems encountered and success achieved." Journal of Information Technology **15**(4): 245-265.

Mason, J. (1996). Qualitative Researching. Thousand Oaks, California, Sage Publications, Inc.

McLean, E. R. (2001). Personal Communication.

Melville, N. and K. Kraemer (2004). "Review: Information Technology And Organizational Performance: An Integrative Model of IT Business Value." MIS Quarterly **28**(2): 283-322.

Miles, M. B. and A. M. Huberman (1994). Qualitative Data Analysis. Thousand Oaks, CA, Sage Publications, Inc.

Minasi, M. and E. Garde (1999, March). "Toward More Reliable Software." WindowsITPro from http://ntmag.com/Articles/Index.cfm?ArticleID=4908.

Mintzberg, H., D. Raisinghani, et al. (1976). "The Structure of 'Unstructured' Decision Processes." Administrative Science Quarterly **21**(2): 246-275.

Mitchell, R. K., B. R. Agle, et al. (1997). "Toward A Theory of Stakeholder Identification and Salience: Defining the principle of Who and What Really Counts." Academy of Management Review **22**(4): 853-886.

Mohr and L. H. (1982). Explaining Organizational Behavior. San Francisco, CA, Jossey-Bass Inc.

Nah, F. F.-H., S. Faja, et al. (2001). "Characteristics of ERP software maintenance: a multiple case study." Journal of Software Maintenance and Evolution: Research and Practice **13**: 399-414.

Nelson, R. R. and P. H. Cheney (1987). "Training End Users: An Exploratory Study." MIS Quarterly **11**(4): 547-560.

Orlikowski, W. J. and J. J. Baroudi (1991). "Studying Information Technology in Organizations: Research Approaches and Assumptions." Information Systems Research **2**(1): 1- 28.

Paine, M. (2000). Making Software Upgrades A First-Class Experience. Health Management. **21:** 22-23.

Pfeffer, J. and G. R. Salancik (1978). The External Control of Organizations A Resource Dependence Perspective. New York, Harper & Row, Publishers.

Robey, D., J. W. Ross, et al. (2002). "Learning to Implement Enterprise Systems: An Exploratory Study of the Dialectics of Change." Journal of Management Information Systems **19**(1): 17-46.

Rogers, E. M. (1983). Diffusion of Innovations. New York, The Free Press, A Division of Macmillan Publishing Co., Inc.

Rowley, T. J. (1997). "Moving Beyond Dyadic Ties: A Network Theory of Stakeholder Influences." Academy of Mangement Review **22**(4): 887-910.

Ryan, S. D. and D. A. Harrison (2000). "Considering Social Subsystem Costs and Benefits in Information Technology Investment Decisions: A View from the Field on Anticipated Payoffs." Journal of Management Information Systems **16**(4): 11-40.

Sauders, C. S. and J. W. Jones (1992). "Measuring Performance of the Information Systems Function." Journal of Management Information Systems **8**(4): 63-82.

Schmidt, R., K. Lyytinen, et al. (2001). "Identifying Software Project Risks: An International Delphi Study." Journal of Management Information Systems **17**(4): 5-36.

Somers, T. M. and K. G. Nelson (2004). "A taxonomy of players and activities across the ERP project life cycle." Information and Management **41**: 257-278.

Stone, E. F. (1991). Research Methods in Organizational Behavior. Glenview, IL, Scott, Foresman and Company.

Strauss, A. and J. Corbin (1990). Basics of Qualitative Research. Newbury Park, Sage

publications.

Swanson, E. B. and C. M. Beath (1989). Maintaining Information Systems in Organizations. Chichester, John Wiley & Sons.

Talley, B. and L. Mitchell (2000). Windows 2000 May Be Unavoidable, But with proper planning you can still sleep soundly. InfoWorld. **22:** 72.

Thatcher, J. B. and P. L. Perrewe` (2002). "An Empirical Examination of Individual Traits As Antecedents to Computer Anxiety and Computer Self-Efficacy." MIS Quarterly **26**(4): 381-396.

Thong, J. Y. L. (1999). "An Integrated Model of Information Systems Adoption in Small Businesses." Journal of Management Information Systems **15**(4): 187-214.

Thong, J. Y. L., C.-S. Yap, et al. (1996). "Top Management Support, External Expertise and Information Systems Implementation in Small Business." Information Systems Research **7**(2): 248-267.

Wallace, L., M. Keil, et al. (2004). "Understanding software project risk: a cluster analysis." Information and Management **42**: 115-125.

Walsham, G. (1995). "The Emergence of Interpretivism in IS Research." Information Systems

Research **6**(4): 376-394.

Watson, R. T., G. G. Kelly, et al. (1997). "Key Issues in Information Systems Management: An International Perspective." Journal of Management Information Systems **13**(4): 91-115.

Weill, P. (1992). "The Relationship Between Investment in Information Technology and Firm Performance: A Study of the Valve Manufacturing Sector." Information Systems Research **3**(4): 307-333.

Weiss, T. R. (2000). ADP Upgrade Glitch Causes Online Trading Problems. computerworld. **34:** 20.

Weston, R. (1997). SAP tries to beef up R/3 manageability. Computerworld. **31:** 14.

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

VITAE

Education

2006	Doctor of Philosophy in Business Administration (CIS), Georgia State University
1995	Master of Business Administration, Mississippi State University
1993	Bachelor of Science in Computer Science, University of Mississippi

Work Experience

2005 –	Assistant Professor, Department of Information Systems and Technology Management, University of Texas, San Antonio
2003 – 2005	Instructor, Department of Information Systems and Technology Management, University of Texas, San Antonio
1998 – 2002	Graduate Teaching and Research Assistant, Georgia State University
1998	Independent Contractor, Intelliquest Communications, Inc., Atlanta, GA
1995 – 1998	Database Specialist / Programmer Analyst II, Information America, Atlanta, GA

Publications

Chua, C., H. M. Khoo, D. W. Straub, S. Kadiyala, D. Kuechler. (In press). “The Evolution of E-commerce Research: A Stakeholder Perspective.” Journal of Electronic Commerce Research

Chua, C., G. Rose, H. M. Khoo, D. W. Straub. (2005). "Technological Impediments to B2C Electronic Commerce: An Update." Communications of the AIS 16(7): 127-216.

Robey, D., H. M. Khoo, and C. Power. (2000) "Situated Learning in Cross-functional Virtual Teams." IEEE Transactions on Professional Communication 43 (1): 51-66; and Technical Communication 47 (1): 51-66. (Joint Special Issue)

Rose, G., H. M. Khoo, D. W. Straub. (1999). "Current Technological Impediments to Consumer Electronic Commerce on the World Wide Web." Communications of the AIS 1(16): 1- 73.

Honors

Graduate Teaching Assistant Teaching Excellence Award, 2002

Southern Regional Educational Board (SREB) Dissertation Year Fellowship, 2002

GSU Dissertation Grant Award, 2002

Listed in Who's Who in International Professional Management, 1999