THE INTERACTION BETWEEN CONTEXT AND TECHNOLOGY DURING
INFORMATION SYSTEMS DEVELOPMENT (ISD)
action research investigations in two health settings

by
MIKE CHIASSON
B.Com., University of Alberta, 1990

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in
THE FACULTY OF GRADUATE STUDIES
THE FACULTY OF COMMERCE AND BUSINESS ADMINISTRATION
Department of Management Information Systems

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA
JULY 1996
©Mike Chiasson, 1996
In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Commerce

The University of British Columbia
Vancouver, Canada

Date July 31, 1996
ABSTRACT

Software development and implementation failure is perceived by developers and users as a serious problem. Of every six new software development projects, 2 are abandoned, the average project lasts 50% longer than expected, and 75% of large systems are “operating failures” that are rejected or perform poorly. Design failure contributes to the productivity paradox, where increased investment in information technology (IT) has not correlated with improvements in productivity. Many IS researchers state that further research examining the interaction between technology and context during information system development (ISD) is required. This current study is motivated by these calls for research.

The marrying of information systems and health research also raises a second motivation. The deployment and diffusion of IT can contribute to the effective utilization of health resources. Another motivation of the thesis is to explore the effect of information systems on disease prevention, and provide an opportunity to develop and diffuse IT tools that promote health.

To address these two motivations, two case studies of ISD in two health studies are described. The first case study involved the initiation and development of an electronic patient record in two outpatient clinics specializing in heart disease prevention and rehabilitation (SoftHeart). The second case study involved the development of a windows-based multimedia software that assists the planning of breast cancer educational and policy programs in communities. The first case study covered four years (Summer of 1992 to Spring of 1996) and the second case covered 1 year (Spring of 1995 to Spring of 1996). The purpose of the thesis is to generate hypotheses for future research in ISD.
Both studies employed an “action research” approach where the researcher was directly involved with software design and programming. Data from interviews, meeting minutes, field notes, design and programming notes, and other documentation were collected from both studies and triangulated to provide valid interpretations. Important and illustrative technology-context events are extracted from the cases to uncover processes between technology and context during stages of development. Processes are compared with four theories linking technology and context: technological imperative and organizational imperative (unidirectional), and emergent perspective and social technology (bi-directional). These processes are then combined to reach tentative conclusions about the ISD process.

Key findings indicate an interplay between a small number unidirectional processes (organizational and technology imperative) and a large number of bi-directional theories (social technology and emergence). Overall, the emergent perspective described or participated in describing a majority of the processes, given the developer’s perspective, extraction and interpretation of these key processes. In both cases, the ISD trajectory was best described as emergent.

The result of within case and cross-case analysis is a model integrating the four technology-context theories depending on stakeholder agreement and the adaptability of technology during development and use. Dynamics and change in task, technology, and stakeholder configurations are explained by the deliberate or accidental interaction of new and old stakeholders, technology, ideas, agreements and/or tasks over time. Implications for research and practice are discussed.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABSTRACT</strong></td>
</tr>
<tr>
<td><strong>TABLE OF CONTENTS</strong></td>
</tr>
<tr>
<td><strong>LIST OF TABLES</strong></td>
</tr>
<tr>
<td><strong>LIST OF FIGURES</strong></td>
</tr>
<tr>
<td><strong>ACKNOWLEDGEMENT</strong></td>
</tr>
</tbody>
</table>

1. **INTRODUCTION** | 1 |
   - **Description of the Thesis** | 1 |
   - **Description of Case Studies and Methodology** | 1 |
   - **Study Motivation** | 2 |
     - **Motivation 1: Information Systems Failure** | 3 |
     - **Motivation 2: Combining ISD and Health** | 5 |
   - **Key Findings** | 6 |
   - **Organization of Thesis** | 6 |

2. **LITERATURE REVIEW** | 8 |
   - **Chapter Overview** | 8 |
   - **Stages and Dimensions of System Development** | 9 |
   - **Technology's Linkage to Contextual Dimensions** | 11 |
   - **Four Theories of the Technology - Context** | 13 |
     - **Technological Imperative** | 15 |
     - **Technological Imperative Research** | 16 |
     - **Organizational Imperative** | 17 |
     - **Organizational Imperative Research** | 19 |
     - **Reflections on Unidirectional Theories of Technology-Context Interaction** | 24 |
     - **Emergent Perspective** | 28 |
     - **Emergent Perspective Research** | 29 |
     - **Social Technology** | 40 |
     - **Social Technology Research** | 42 |
   - **Conclusion: Reconciling the Four Theories of Technology and Context Interaction** | 45 |
   - **Chapter Summary** | 46 |

3. **RESEARCH QUESTION AND METHODOLOGY** | 48 |
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2-1</td>
<td>Stages and Dimensions of System Development</td>
<td>9</td>
</tr>
<tr>
<td>Table 2-2</td>
<td>Actions and Products of Development Stages - Cooper &amp; Zmud (1990)</td>
<td>10</td>
</tr>
<tr>
<td>Table 2-3</td>
<td>Revised Technological and Contextual Dimensions of System Development</td>
<td>11</td>
</tr>
<tr>
<td>Table 2-4</td>
<td>Process Theories of the Technology and Contextual Dimensions Linkage</td>
<td>14</td>
</tr>
<tr>
<td>Table 2-5</td>
<td>Reconciling the Four Theories</td>
<td>46</td>
</tr>
<tr>
<td>Table 2-6</td>
<td>Summary of Four Process Theories and ISD Research</td>
<td>47</td>
</tr>
<tr>
<td>Table 3-1</td>
<td>Development Environment</td>
<td>51</td>
</tr>
<tr>
<td>Table 3-2</td>
<td>Characteristics of a Process</td>
<td>60</td>
</tr>
<tr>
<td>Table 4-1</td>
<td>Process Stages in the Development of SoftHeart</td>
<td>64</td>
</tr>
<tr>
<td>Table 4-2</td>
<td>Processes in Project Emergence</td>
<td>74</td>
</tr>
<tr>
<td>Table 4-3</td>
<td>Processes in Health Funders Skeptical</td>
<td>76</td>
</tr>
<tr>
<td>Table 4-4</td>
<td>Processes in Our Response to Health Critiques</td>
<td>78</td>
</tr>
<tr>
<td>Table 4-5</td>
<td>Processes in the Environmental Shocks Stage</td>
<td>80</td>
</tr>
<tr>
<td>Table 4-6</td>
<td>Processes in the Move Ahead Stage</td>
<td>83</td>
</tr>
<tr>
<td>Table 4-7</td>
<td>Processes in the Interest and Constraint Stage</td>
<td>87</td>
</tr>
<tr>
<td>Table 4-8</td>
<td>Processes in the Conflict Stage</td>
<td>90</td>
</tr>
<tr>
<td>Table 4-9</td>
<td>Processes in the Decision to Act Stage</td>
<td>92</td>
</tr>
<tr>
<td>Table 4-10</td>
<td>Processes in the Momentum Stage</td>
<td>94</td>
</tr>
<tr>
<td>Table 4-11</td>
<td>Processes in the Development Quagmire Stage</td>
<td>96</td>
</tr>
<tr>
<td>Table 4-12</td>
<td>Processes in the Recovery Stage</td>
<td>98</td>
</tr>
<tr>
<td>Table 4-13</td>
<td>Processes in the Future of SoftHeart</td>
<td>100</td>
</tr>
<tr>
<td>Table 4-14</td>
<td>Summary of Process Types in Process Stages</td>
<td>101</td>
</tr>
<tr>
<td>Table 4-15</td>
<td>ISD Research Categories within Process Stages</td>
<td>102</td>
</tr>
<tr>
<td>Table 4-16</td>
<td>Examination of Shared Processes</td>
<td>104</td>
</tr>
<tr>
<td>Table 4-17</td>
<td>Number of Times One Process Type is Followed Another Type</td>
<td>105</td>
</tr>
</tbody>
</table>
TABLE 4-18: NUMBER OF TIMES A SERIES OF ONE PROCESS TYPE WAS TERMINATED BY ANOTHER TYPE ........106

TABLE 5-1: PROCESS STAGES IN THE DEVELOPMENT OF EMPOWER ......................................................... 108

TABLE 5-2: PROCESSES IN THE EMERGENCE STAGE ............................................................................. 115

TABLE 5-3: PROCESSES IN THE "GETTING READY" STAGE ................................................................. 118

TABLE 5-4: PROCESSES IN THE "PANDORA'S BOX OPENS" STAGE ................................................... 124

TABLE 5-5: PROCESSES IN THE "SQUEEZING THE MOST" STAGE ..................................................... 127

TABLE 5-6: PROCESSES IN "VISIONS OF A NEW EMPOWER" STAGE .............................................. 128

TABLE 5-7: SUMMARY OF PROCESS TYPES IN PROCESS STAGES ..................................................... 129

TABLE 5-8: ISD RESEARCH CATEGORIES WITHIN PROCESS STAGES .............................................. 130

TABLE 5-9: EXAMINATION OF SHARED PROCESSES ......................................................................... 131

TABLE 5-10: NUMBER OF TIMES A PROCESS TYPE WAS FOLLOWED BY ANOTHER TYPE .................. 132

TABLE 5-11: NUMBER OF TIMES A SERIES OF ONE TYPE WAS TERMINATED BY ANOTHER TYPE .... 132

TABLE 6-1: TECHNOLOGICAL IMPERATIVE SUMMARY FROM CASE ONE AND CASE TWO ............. 135

TABLE 6-2: ORGANIZATIONAL IMPERATIVE SUMMARY FROM CASE ONE AND CASE TWO ............ 139

TABLE 6-3: EMERGENT PERSPECTIVE SUMMARY FROM CASE ONE AND CASE TWO ......................... 143

TABLE 6-4: SOCIAL TECHNOLOGY SUMMARY FROM CASE ONE AND CASE TWO ......................... 148
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>ACTION AND ACTION FRONTIER</td>
<td>12</td>
</tr>
<tr>
<td>2-2</td>
<td>TECHNOLOGICAL IMPERATIVE</td>
<td>15</td>
</tr>
<tr>
<td>2-3</td>
<td>THE ORGANIZATIONAL IMPERATIVE</td>
<td>19</td>
</tr>
<tr>
<td>2-4</td>
<td>THE EMERGENT PERSPECTIVE ON TECHNOLOGY AND CONTEXT</td>
<td>28</td>
</tr>
<tr>
<td>2-5</td>
<td>SOCIAL TECHNOLOGY THEORY</td>
<td>41</td>
</tr>
<tr>
<td>3-1</td>
<td>RESEARCH DESIGN</td>
<td>50</td>
</tr>
<tr>
<td>3-2</td>
<td>LADDER OF INFERENCE</td>
<td>59</td>
</tr>
<tr>
<td>3-3</td>
<td>LEVELS OF TECHNOLOGY</td>
<td>61</td>
</tr>
<tr>
<td>4-1</td>
<td>DATA VIEW OF SOFTHEART</td>
<td>69</td>
</tr>
<tr>
<td>5-1</td>
<td>EMPOWER PROGRAM STRUCTURE</td>
<td>111</td>
</tr>
<tr>
<td>6-1</td>
<td>INTEGRATED MODEL OF CONTEXT AND TECHNOLOGY INTERACTION</td>
<td>154</td>
</tr>
<tr>
<td>D-1</td>
<td>PRECEDE/PROCEED MODEL</td>
<td>261</td>
</tr>
<tr>
<td>E-1</td>
<td>INTRODUCTORY SCREEN</td>
<td>266</td>
</tr>
<tr>
<td>E-2</td>
<td>PATIENT SEARCH WINDOW</td>
<td>267</td>
</tr>
<tr>
<td>E-3</td>
<td>PATIENT INFORMATION PANEL</td>
<td>268</td>
</tr>
<tr>
<td>E-4</td>
<td>LIPIDS INFORMATION PANEL</td>
<td>269</td>
</tr>
<tr>
<td>E-5</td>
<td>MEDICATIONS INFORMATION PANEL</td>
<td>270</td>
</tr>
<tr>
<td>E-6</td>
<td>LIPIDS EDIT PANEL</td>
<td>271</td>
</tr>
<tr>
<td>E-7</td>
<td>PATIENT INFORMATION EDIT PANEL</td>
<td>272</td>
</tr>
<tr>
<td>E-8</td>
<td>TRENDS INFORMATION PANEL</td>
<td>273</td>
</tr>
<tr>
<td>E-9</td>
<td>SCHEDULING PANEL</td>
<td>274</td>
</tr>
<tr>
<td>E-10</td>
<td>REPORTING PANEL</td>
<td>275</td>
</tr>
<tr>
<td>F-1</td>
<td>INTRODUCTORY SCREEN</td>
<td>276</td>
</tr>
<tr>
<td>F-2</td>
<td>ASSUMPTIONS SCREEN</td>
<td>277</td>
</tr>
<tr>
<td>F-3</td>
<td>GLOBAL MAP PAGE</td>
<td>278</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENT

This thesis demonstrates the importance of “context” in producing invention. I would especially like to thank many individuals around me who have provided substantial and continuous support for the project. I owe a great debt of gratitude to my supervisor, professor Albert Dexter, who provided time, encouragement and freedom to explore new ideas. I also owe a great debt to my two committee members: professors Maurice Levi and Lawrence Green. Both have provided me with precious time and guidance in producing this thesis. I am grateful to the two organizations who supported my research, who allowed me to experiment with their information systems and organizational operations, and assisted me in faithfully reporting the findings. I am indebted to my mother, Beth Chiasson, who provided me with an early thirst for learning and understanding, and provided both intellectual and emotional support to my sociological and methodological ideas and references. I owe great thanks to Laurenda Daniells, who provided a home, detailed insights into the university process, important information for surviving in my new city, and wonderful discussion and friendship throughout my graduate years. I also give my warmest appreciation to those colleagues who challenged my thinking, provided wonderful hours of discussion and laughter, and contributed greatly to the thesis: Amos Shachar, Janice Foley, Gary Clark, and Dawne Milligan. I am grateful for the family and friends who continuously and unconditionally supported me with discussion, listening, and laughter: my father Gerry Chiasson, my lovely girlfriend Frances Brodie, my two brothers David and Larry Chiasson, and my faithful companion Gracie.
1. **Introduction**

1.1. **Description of the Thesis**

*Information systems development* (ISD) involves a *process*\(^1\) of interaction between *technology*\(^2\) and *context*\(^3\) that results in the initiation, creation, and early implementation of an information system (IS) and information technology (IT). Most of the ISD literature is concerned with descriptive (what is) or normative (what should be) claims about the technology-context interaction. This thesis explores information systems development (ISD) in two health settings, generating hypotheses about the ISD process. The following research question is addressed:

   (1) What is the nature of the interaction between technology and context during ISD?

1.2. **Description of Case Studies and Methodology**

The first case study describes the initiation and development of an electronic patient record in two outpatient clinics specializing in heart disease prevention and rehabilitation (SoftHeart). The second case study describes the development of a windows-based multimedia software that assists the planning of breast cancer educational and policy programs in communities (EMPOWER - Expert Methods of Planning and Organizing Within

---

\(^1\) *Process* of interaction between technology and context involves time, an “exchange”, and elements from technology and context that contribute in the exchange.

\(^2\) *Technology* is a general term in this thesis, used to describe all aspects of the information system including software and infrastructure. It is also used to describe other equipment, machinery, and/or administrative processes in non-IS research.

\(^3\) *Context* include: users, organization, environment, task, developers/suppliers, and environment.
Chapter 1: Introduction

Everyone's Reach. The first case study covered four years (Summer of 1992 to Spring of 1996) and the second case covered 1 year (Spring of 1995 to Spring of 1996). The purpose of the thesis is to generate hypotheses for future research in ISD.

Both studies employed an "action research" approach where the researcher was directly involved with software design and programming. The action research method allows the researcher to address both descriptive and normative questions about ISD. Data from interviews, meeting minutes, field notes, design and programming notes, and other documentation were collected from both studies and triangulated to provide valid interpretations. Important and illustrative events are extracted from the cases to uncover processes between technology and context during stages of development (Newman & Robey 1992). Processes are compared with four theories linking technology and context: technological imperative and organizational imperative (unidirectional), and emergent perspective and social technology (bi-directional). These processes are then combined to reach tentative conclusions about the ISD process.

1.3. Study motivation

This thesis is motivated by important issues in information systems research and medicine/health promotion.

---

4 Discrete points in time when technology shapes or is shaped by context (users, developers, organization, environment, task)

5 Certain periods of time where development carries on a certain general form of interaction between technology and context (e.g. initiation and creation, development, conflict, etc.).
1.3.1. Motivation 1: Information Systems Failure

Software development and implementation failure is perceived by developers and users as a serious problem (Lyytinen 1987; Gibbs 1994; Landauer 1995). Of every six new software development projects, 2 are abandoned, the average project lasts 50% longer than expected, and 75% of large systems are “operating failures” that are rejected or perform poorly (Gibbs 1994). Design failure contributes to the productivity paradox, where increased investment in information technology (IT) has not correlated with improvements in productivity (Burton 1992; Landauer 1995).6

There is disagreement in the IS community, however, over the cause and solutions to these development problems. Some recommend more strategic planning and “linking” of the information system (IS) and information technology (IT) plans with business strategy (McFarlan 1984; Earl 1989). Some recommend the use of better formalized methods and tools to model business environments (Coad & Yourdon 1991; Wand & Weber 1993). Other researchers recommend the use of user involvement and testing, and more humanistic and interpretive approaches to analyzing decision makers (Boland 1985; Boland 1987; Landauer 1995). Still others fault developers for focusing only on technology while ignoring social context (Kling 1980; Kling & Scacchi 1982; Kling 1987; Turner 1987). Finally, some researchers comment that the social context around technology appropriates and shapes the effect of technology within certain boundaries (Orlikowski 1991, 1992a; 1992b; Zuboff 1988).

Each view carries implicit assumptions about the relative influence of context on technology,

---

6 The literature around the “productivity paradox” is controversial and beyond the scope of this thesis. See Hitt & Brynjolfsson (1994) and Landauer (1995) for a more complete discussion.
and technology on context during information systems development (ISD). For example, strategic planning approaches carry implicit diffusion and demand notions of system development. Specifically critiquing the implicit demand and diffusion models of ISD, Attewell (1992) quotes Eveland & Tornatzky’s (1990) who remark:

Problems arise when the diffusion model is applied in situations where its basic assumptions are not met -- that is to say, virtually every case involving complex, advanced technology (p. 123).

Further research examining the interaction between technology and context to understand the process of ISD is required. Several researchers have commented on the need for research to understand the process of ISD and technology innovation more generally (Turner 1987; Markus & Robey 1988; Cooper & Zmud 1990; Eveland & Tornatzky 1990; Newman & Robey 1992). Cooper and Zmud (1990) state that very few studies have addressed context (environment, user, developers and suppliers, and organization) and its interaction with technology during ISD, and that little research has attempted to address a number of development stages and contextual dimensions simultaneously:

...that prior studies focus on too few of the model stages and factors ..., and advocate that future research should explore the impact of multiple contextual factors on multiple implementation stages (p. 125).

...studies examining both rational and political forces throughout the IT implementation stages can be especially fruitful. Such studies will be more powerful if they are longitudinal in nature -- providing the capability to more thoroughly examine the dynamics of individual, organizational, and technological adaptations across the implementation stages (p. 137).

Markus & Robey (1988) also recommend further process research mixing levels of analysis (individual, group, organization, etc.). Process theories are considered more appropriate for studying complex social phenomena:
Chapter 1: Introduction

In summary, we believe that process theories are useful precisely because, while recognizing and accepting the complexity of causal relationships, they do not abandon the goals of generalizability and prediction....With care, findings can be generalized to other settings, and predictions can be tested in later research. The advantage of process theories is that these predictions may correspond more faithfully to actual events in organizations than do the typical predictions of variance formulations (p. 593).

This current study is motivated by these calls for research. The purpose of this thesis is to reconcile the ISD literature, and provide empirical data on the process of ISD.

1.3.2. Motivation 2: Combining ISD and Health

The marrying of the information systems and health addresses the first motivation and raises a second motivation. With respect to the first motivation, the health setting provides a complex and unique environment to test and expand our understanding of IS development (Kling 1980). With respect to a second motivation, the deployment and diffusion of IT can contribute to the effective utilization of health resources. The research explores the effect of information systems on disease prevention, and provides an opportunity to develop and diffuse IT tools that promote health.

7 The Canadian health sector has a number of complex and turbulent environmental characteristics that influence IS development (Emery & Trist 1971). Funding arrangements vary widely from not-for-profit vs. profit, fee-for-service vs. payment by capitation, research versus clinical, etc. The labor force is composed of numerous semi-autonomous professionals providing skilled services. Information and decision making involves both quantitative and qualitative data. Measurement of health outcomes is very difficult to determine at both the individual and organizational level of analysis. There are few standardized information systems compared with other industries. Finally, public debate, input and involvement in health care is extensive. In this sense, the setting provides an outstanding opportunity to test and expand our knowledge of factors, processes, and events influencing information systems development. See Appendix C for a more detailed discussion.

8 As of 1990, health care expenditures comprise 9% and 12.5% of the GNP in Canada and the United States respectively (Evans 1990). There is extensive literature and press coverage suggesting that a significant portion of these resources are not being used effectively, and opportunities exist for researching alternative technologies and delivery approaches.

9 Although this thesis does not provide direct empirical evidence of the IT/IS and health outcomes link, a sampling of the health literature linking IT/IS to processes affecting health outcomes is discussed in the Appendices.
1.4. Key Findings

Four process theories of context-technology interaction are used in this study. Two unidirectional theories (technological imperative and organizational imperative) and two bi-directional theories (emergent perspective and social technology) are used to organize the literature and the case data. Technology-context processes were categorized by level and technology part, the time period, the stakeholders involved, the tasks, and one or more of the four process theories that best described the process.

Key findings indicate an interplay between a small number unidirectional processes (organizational and technology imperative) and a large number of bi-directional theories (social technology and emergence). Overall, the emergent perspective described or participated in describing a majority of the processes, given the developer's perspective, extraction and interpretation of key processes. In both cases, the ISD trajectory was best described as emergent.

The result of within case and cross-case analysis is a model integrating the four technology-context theories depending on stakeholder agreement and the adaptability of technology during development and use. Dynamics and change in task, technology, and stakeholder configurations are explained by the deliberate or accidental interaction of new and old stakeholders, technology, ideas, agreements and/or tasks over time. Implications for research and practice are discussed.

1.5. Organization of Thesis

The thesis is organized as follows. Chapter 2 provides a review of the Information Systems Development (ISD) literature. Process research and theories of ISD and technology
Chapter 1: Introduction

are discussed and integrated to provide a framework and language for interpreting the case data. Chapter 3 discusses the action research methodology employed in the two case studies, providing an understanding of the method's strengths and weaknesses in addressing the research question. Chapter 4 describes the interaction between context and technology during the development of SoftHeart over a four-year period, providing a substantive theory of the SoftHeart case. Chapter 5 describes the interaction between context and technology during the development of EMPOWER over a twelve-month period, providing a substantive theory of the EMPOWER case. Chapter 6 compares and contrasts the two case studies and relates the findings to the literature. The chapter concludes with a model integrating task, technology, stakeholders, and the four process theories. Chapter 7 draws implications, describes weaknesses, and suggests directions for future research.\textsuperscript{10}

\textsuperscript{10} Appendices are provided to support the cases including case descriptions for SoftHeart (Appendix A) and Empower (Appendix B), medical and health promotion literature important in case one (Appendix C) and case two (Appendix D), and snapshots of the screens from SoftHeart (Appendix E) and Empower (Appendix F). Appendix G provides a summary of the SoftHeart program structure. Finally, Appendix H provides an example and a summary of the "rules" used to choose the four technology-context theories, categorize the IS and technology development literature into the four theories, analyze event and process construction, and attach processes to the literature.
2. Literature Review

2.1. Chapter Overview

The information systems development (ISD) literature holds various descriptive (what is) and normative (what should be) conclusions about the technology-context interaction. Chapter 2 develops a framework and language to understand and integrate these various views, providing alternative theories and perspectives for viewing the case data.

ISD research is a fairly large and widespread body of research. However, there are certain characteristics of the ISD research that are particularly applicable to this thesis. First, longitudinal and process-focused research in field settings, with a level of analysis at a group or organizational level, are more relevant to this thesis than variance research. Second, the literature review emphasizes three areas of ISD research. Using the keyword classification of Barki, Rivard, & Tabot (1993), they include IS Implementation, organizational dynamics, and information systems project management.

The chapter is organized as follows. First, the chapter identifies the stages and dimensions of system development and implementation. Second, the chapter introduces the “action” and “action frontier” concepts to assist in examining the interaction between the technology and the contextual dimensions. Third, four process theories (two unidirectional and two bi-directional) are examined that link technology to context: the technological imperative, organizational imperative, emergent perspective, and social technology are examined. Classic and exemplar ISD research is discussed under each theory to provide
evidence, normative conclusions, and critiques of each of the four theories. This provides a rich understanding of the ISD process (Franz & Robey 1984; Hirschheim & Klein 1989; Orlikowski & Baroudi 1991; Orlikowski & Robey 1991; Newman & Robey 1992). The chapter concludes with methods of comparing and reconciling these four theories, providing a basis for building an integrated theoretical framework.

2.2. Stages and Dimensions of System Development

Cooper and Zmud’s (1990) framework of the IS implementation literature provides a foundation for this study’s framework. Their meta-analysis classifies the IS implementation research into six stages: initiation (unfreezing), adoption and adaptation (changing), and acceptance, routinization, and infusion (refreezing). They also identify five dimensions that affect and/or are affected by the activity of each stage: user, organization, task, technology, and environment. They construct the following table to show where previous research has focused:

<table>
<thead>
<tr>
<th>Stage/Dimension</th>
<th>User</th>
<th>Organization</th>
<th>Task</th>
<th>Technology</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adoption</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Adaptation</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Acceptance</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Routinization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Infusion</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There are both actions and products for each of the implementation stages:
Table 2-2: Actions and Products of Development Stages - Cooper & Zmud (1990)

<table>
<thead>
<tr>
<th>Stage</th>
<th>Actions</th>
<th>Products</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiation</td>
<td>Search for potential solutions to a problem.</td>
<td>Match of potential solution with a problem.</td>
</tr>
<tr>
<td>Adoption</td>
<td>Rational and political discussions gain organizational backing for the project.</td>
<td>Commitment of resources and time.</td>
</tr>
<tr>
<td>Adaptation</td>
<td>System is developed, installed and maintained.</td>
<td>A working set of IT.</td>
</tr>
<tr>
<td>Acceptance</td>
<td>Employees willing to commit time to the application.</td>
<td>Widespread use of IT throughout the organization.</td>
</tr>
<tr>
<td>Routinization</td>
<td>IT becomes a normal part of operations.</td>
<td>Transparent IT that is embedded in the organization.</td>
</tr>
<tr>
<td>Infusion</td>
<td>IT enhances effectiveness and the IT begins to support higher level tasks.</td>
<td>Its product is an IT used to its fullest potential.</td>
</tr>
</tbody>
</table>

These stages can be seen as overlapping, often with some elements of a new software system reaching the infusion stage before other parts of the system are created.

Cooper and Zmud also identify dimensions involved in IS development and implementation. These include technology, and contextual dimensions surrounding the technology including user, organization, task and environment. Various characteristics and “stakeholders” of these contextual dimensions affect and are affected by ISD. To this framework is added the “developers and suppliers” dimension from the supply-side discussion of ISD (Attewell 1992). The revised framework appears as follows:
2.3. Technology's linkage to Contextual Dimensions

The previous section provided a framework for describing and categorizing development stages and dimensions. However, theories about the interaction between the dimensions are missing. Of particular importance is the linkage between technology, which includes information systems (IS) and information technology (IT), and the other “contextual dimensions”, which include task, user, organization, environment, and developers and suppliers of technology.

To assist with the interpretation of the various theories, the dimensions are collapsed into “action” and the “action frontier”. Task and organization will be represented as a single construct called “action”. Technology, user, developers and suppliers, and environment are the boundaries and constraining factors around task and organization, and the overlaps among these five dimensions will be called the “action frontier”. The action frontier establishes the boundary of possible tasks and organizations. This implies that when examining information systems, we are not as concerned with the technology itself as with the “enabling” qualities
that the technology provides to the organization.\textsuperscript{11}

The following diagram illustrates these two new constructs:

**Figure 2-1: Action and Action Frontier**

The "action frontier" sets the boundaries around the organization and its tasks; what it can do. "Action" is the actual tasks and organizational characteristics of an organization. Using the above conceptual diagram will help with the interpretation of technology and context theories.

\textsuperscript{11} This is somewhat similar to the socio-technical view of technology, suggesting that technology and organizational/task structure are matched and result in certain outcomes.
2.4. **Four Theories of the Technology - Context**

In this section, four process theories of technology's linkage to context are described. The purpose of the theories is to provide a set of views of technology and context with which to describe and organize the literature and case data. Several key articles and books discuss the linkage between technology and the stakeholders of contextual dimensions, and identify four theories of technology-context interaction (Kling 1980\(^{12}\); Kling & Scacchi 1982\(^{13}\); Markus 1983\(^{14}\); Van De Ven 1986\(^{15}\); Markus & Robey 1988\(^{16}\); Hirschheim & Klein 1989\(^{17}\); Burton 1992\(^{18}\); DeSanctis & Poole 1994\(^{19}\); Myers 1994b\(^{20}\); Thomas 1994\(^{21}\)). The four theories are the *technological imperative, organizational imperative, emergent perspective,* and *social technology.* A summary of each theory is included below, providing a set of inference rules for categorizing the literature into the four theories:

---

12 Kling discusses *systems rationalism* and *segmented institutionalism.*
13 Kling and Scacchi discuss *discrete entity* and *web models* of computing.
14 Markus' theories include *system driven, people driven,* and two interaction theories: *political* and *socio-technical.*
15 Van De Ven's theories include *technology-driven,* *customer or need-driven,* and *simultaneous coupling.*
16 Markus & Robey include *technological imperative,* *organizational imperative,* and *emergent perspective.*
17 Hirschheim & Klein focus on the approach and assumptions of the developer toward the organization including *functionalist,* *integrationist,* *radical structuralist,* and *neo-humanist.*
18 Burton critically analyzes *technological determinism* and *societal determinism.*
19 DeSanctis & Poole include *decision making,* *institutional school,* and *social technology.*
20 Myers includes the *technology acceptance,* *organizational change,* and *organizational problem-solving involving mutual adaptation.*
21 Thomas includes *technological determinism,* *social choice,* and *power-process.*
Table 2-4: Process Theories of the Technology and Contextual Dimensions Linkage

<table>
<thead>
<tr>
<th>Perspective /Characteristics</th>
<th>Direction of Causality</th>
<th>Main Assumption</th>
<th>Role of Users and Developers/Suppliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technological Imperative Theory (TI)</td>
<td>Technology to Context</td>
<td>Technology carries impact within itself</td>
<td>Developers have a simple role of technology installation. In moderate form, users can choose and use the technology. In extreme form, users have no choice.</td>
</tr>
<tr>
<td>Organizational Imperative Theory (OI)</td>
<td>Context to Technology</td>
<td>Context rationally chooses, uses, and/or develops technology to suit its needs.</td>
<td>Designers implement technology identified by top management and/or integrated views of users.</td>
</tr>
<tr>
<td>Emergent Perspective Theory (EP)</td>
<td>Causality is complex between context and technology.</td>
<td>Technology is composed of parts that carry social meaning. The result of context-technology interaction is explainable but unpredictable.</td>
<td>Developers’ and users’ ideas and visions determine the criteria for selection, development approaches, the choice, the choices within the choice, and use.</td>
</tr>
<tr>
<td>Social Technology Theory (ST)</td>
<td>Social practices moderate the effect of technology on context.</td>
<td>Both context and technology act as constraints and/or enablers</td>
<td>Developers build technology within constraints and opportunities. Users adapt the technology within technological, environmental, and user constraints</td>
</tr>
</tbody>
</table>

ISD research can be classified based on the researcher’s implicit or explicit process theory and the stages and contextual dimensions studied. My purpose is to demonstrate how the framework and four process theories can be used to understand and integrate the ISD research.

---

22 Most of the research cannot be strictly categorized into one theoretical area because the four theories are linked, and some research use conflicting theories.
2.4.1. Technological Imperative

The *technological imperative* (TI) process theory states that technology impacts context in a unidirectional manner. In its strong form, TI claims that the action frontier is solely determined by technology, and that new technology external to the organization revolutionizes the environment. This revolution forces or allows the organization to adopt the technology and the new task configurations embedded in the technology. Developers have little role in implementing the technology, and users adopt the technology in a predetermined and uniform manner.

The following figure illustrates the technological imperative theory:

**Figure 2-2: Technological Imperative**

[Diagram showing the technological imperative theory with labeled old and new action frontiers, time axis, and process stages]
As the diagram demonstrates, the action frontier is changed by technology, and the organization must adopt the technology to align its tasks within the new action frontier. The organization is forced into mapping itself onto the new action frontier in order to survive.

2.4.2. Technological Imperative Research

Much popular press coverage employs a strong technological imperative and a large percentage of information systems literature employs a moderate technological imperative. There is generally a future-oriented view of the effects of technology on the industry and society. Some focus on the impact of hardware such as information technology (IT) (Mohr 1987\textsuperscript{23}; Gurbaxani & Whang 1991), the "internet"\textsuperscript{24} (Rheingold 1993) computers (Franklin 1990; Postman 1992; Talbott 1995), telecommunications (Hammer & Champy 1993), and computer networks (Sproull & Kiesler 1991).\textsuperscript{25} Some researchers focus on the impact of systems including group decision support systems\textsuperscript{26}, interorganizational systems\textsuperscript{27}, expert

\textsuperscript{23} Mohr (1987) claims that the computer has diffused into many spheres of life, overcoming behavioral inertia. Why? Because the computer can be used and integrated into routine change and readaptation including: search, imitation, craft modernization, market survival, slack distribution, status maintenance, recruitment, play, maturation-socialization, and constituency-satisfaction. Because the computer is seen as a homogeneous tool and solution to many of these issues, it is adopted and adapted easily. The article encompasses a number of principles from all four process theories.

\textsuperscript{24} The internet is a short word for "inter networking", a term used to identify a growing computer communications system of networks linked together by a standard data communications protocol. This allows the passing of electronic messages and other data to other users on the worldwide internet, and the use of web browsers to see information on the "world wide web".

\textsuperscript{25} In fairness to Franklin (1990) and Talbott (1995), they both suggest that it is the technological imperative that creates passivity amongst citizens and users of technology, disempowering them to take control of the technology's future. This suggests that the loss of control over technology can be regained through asserting an organizational or social imperative.

\textsuperscript{26} Group Decisions Support Systems use a set of networked computers in a meeting room to structure and store information about a meeting into the computer. These tools help meeting participants to independently brainstorm ideas, set criteria for selection, reach consensus, vote, and select action steps and solutions.

\textsuperscript{27} The most well-known interorganizational system is the Electronic Funds Transfer system used in Automated Teller Machines (ATM).
Chapter 2: Literature Review

systems\textsuperscript{28}, and executive information systems\textsuperscript{29}, (Johnston & Vitale 1988; Sviokla 1989; George, Easton, Nunamaker & Northcraft 1990; Mohan, Holstein & Adams 1990). It is important to note that many of these studies do mention context’s effect on technology, recognizing the sometimes unexpected outcomes of technological implementation. However, context is classified into categories, technology still plays the dominant role, and the underlying benefits of technology will be realized when context meets the technology’s requirements.

A technological imperative is recognized when an author or researcher places the emphasis on a technology’s uniform and usually positive effect on context. As a result, technology is an object that can be described and separated from its context. This process is called “reification” (Berger & Luckman 1967; Ritzer 1992).\textsuperscript{30} In a stronger form, TI claims the context is irrelevant. This implies that technology is an object whose form and function (Rogers 1995) can be described and separated from its context. Thus, both the physical characteristics and the effect of technology on context is universal.

2.4.3. Organizational Imperative

The organizational imperative (OI) theory describes or recommends the use of a rational planning process to choose the technologies that will meet an organization’s strategic objectives. The organizational imperative claims that the organization determines a new

\textsuperscript{28} Expert Systems are a practical application of artificial intelligence, where the computer stores rules about a well-defined domain of knowledge that can then be used to reach conclusions and suggestions for users based on an initial set of variables.

\textsuperscript{29} EIS is a decision tool using graphical interfaces to summarize data, and “drill down” capabilities to see the details of the summary information.

\textsuperscript{30} Berge & Luckman define it as “the apprehension of human phenomena as if they were things, that is, in nonhuman or possibly supra-human terms” (p. 89).
action configuration, purchases or develops technology to transform the action frontier, and then moves the organization into this new frontier. Like the technological imperative, technology is still viewed as having deterministic effects on the action domain, and the organization and tasks will almost completely overlap the action frontier. The organizational imperative also views the action frontier as being composed largely of technology. Unlike the technological imperative, the organizational imperative claims organizations use rational choice to choose and/or shape the technology.\textsuperscript{31} The role of developers is to implement technology chosen by top management, and the role of users is to adopt the technology according to top management plans. If there are problems with technology development and implementation, it is because of low resources and/or uncertainty about the environment (Kling 1980).

The following figure illustrates the organizational imperative theory:

\textsuperscript{31} However, we will see a number of references from the technological imperative section in the organizational imperative literature.
As the diagram illustrates, the organization chooses or builds technology that allows it to move into a chosen action frontier. Technology is rationally chosen by the business executives. The mapping between organization and action is almost one-to-one.

### 2.4.4. Organizational Imperative Research

Extensive information systems literature displays an organizational imperative. Strategic information systems planning (SISP), system design and analysis (SDA), task/technology fit (TTF), and integrative and ‘human relation’ (Kling 1980; Hirschheim & Klein 1989) forms of participative design and user involvement (PD&UI) are four research areas employing an organizational imperative. SISP and SDA focus on the environment task
and organizational dimensions. PD&UI focuses on the user, developer/supplier, and technology link. SISP and SDA are more top-down approaches to information systems development while integrative and functionalist forms of PD&UI are bottom-up approaches.

*Strategic Information Systems Planning (SISP)* research provides planning methods to harness information systems and technology to achieve business advantage in the industry (McKenney & McFarlan 1982; McFarlan, & McKenney 1983; McFarlan, McKenney, & Pyburn 1983; McFarlan 1984; Earl 1989). Case studies of specific firms achieving or losing competitive advantage through the use of IS and IT are used as evidence of technology's role in achieving competitive advantage (Johnston & Vitale 1988; Main & Short 1989; Kim & Michelman 1990; Reich & Benbasat 1990). In this sense, SISP posits an organizational imperative theory linking technology to context. Organizations use the technology to achieve competitive advantage to make higher profits. IT is viewed as a powerful tool for achieving strategic goals, and IS and IT are either fully shaped or picked by the organization because of inherent capabilities. In most cases, little is mentioned in the SISP literature about important stages of ISD including idea generation, prototyping and implementation.

The key to achieving the competitive advantage is broad IS management and planning (McFarlan, McKenney & Pyburn 1983; Earl 1989). Planning here tends to follow a rational and top-down approach from business strategy to information systems planning. The result is a "linkage"32 between the business strategic plan and the IT/IS strategic plan in the deployment of information systems and information technology to achieve business objectives.

---

32 "Linkage" is a term used to describe the fit between information technology, information systems, and the goals of the organization.
System Design and Analysis (SDA) is a close partner with SISP, and includes processes and tools used by organizations to develop systems identified by strategic planning. Thus, SDA and SISP coexist together in comfort. Examples of SDA include business process reengineering and structured information systems development.

Business Process Reengineering (BPR) promotes the importance of organization, environment, and/or task, but suggests that redesign should follow a prescriptive path: organizational structure needs to change to realize the full potential of the technology (Davenport & Short 1990; Hammer & Champy 1993).

The fundamental error that most companies commit when they look at technology is to view it through the lens of their existing processes. They ask, ‘How can we use these new technological capabilities to enhance or streamline or improve what we are already doing?’ Instead, they should be asking, ‘How can we use technology to allow us to do things that we are not already doing?’ Reengineering, unlike automation, is about innovation. It is about exploiting the latest capabilities of technology to achieve entirely new goals. (Hammer & Champy 1993, p 85).

Structured Information Systems Development (SISD) employs an engineering approach to system development, using methods, tools, and techniques as “social technologies” that are promoted as essential for developing successful information systems. Entity Relationship (ER) diagrams\(^{33}\), Object Oriented Design\(^{34}\), and Data Flow Diagrams\(^{35}\) are

---

\(^{33}\) A diagrammatic technique that is used to model business data that allows easier generation of database designs (especially the relational database design).

\(^{34}\) A diagrammatic technique that models objects, object attributes, methods (i.e. actions that can be carried out by an object), and message communication between objects. The result is a diagram that can be easily programmed in object oriented programming languages.

\(^{35}\) Used as a counterpart with Entity Relationship Diagrams, this diagramming technique models the flow of data and the transformation of that data in processes. Using a hierarchical decomposition, these processes are further developed into subprocesses that exchange data until the lowest level process is mapped. The result is a technique that can easily be programmed into the traditional structured programming languages.
all examples of social technologies used in information systems development (Coad & Yourdon 1991; Kendall & Kendall 1992).  

*Task Technology Fit (TTF)* research addresses the need to design the technology to fit a certain task and/or setting. Most of this research is descriptive, with studies of differences between performance when technology does and does not fit the task, and with prescriptive messages recommending a need to design technology for the task. Task/technology fit evaluates the fit by examining objective characteristics of the task and technology. Two areas of task/technology fit include *system/task* fit, and *information/task* fit (DeLone & McLean 1992).

*System/Task* fit claims that certain tasks and production processes require certain information systems. For example, Cooper and Zmud (1990) examine the system characteristics of Material Requirements Planning (MRP) systems and their adoption and infusion amongst organizations with more versus less complex production planning and control tasks. More complex planning and control tasks were assumed to violate the assumptions of MRP which is more appropriate for continuous production. Cooper & Zmud's (1990) results confirmed that adoption of MRP was decreased by more complex production planning and control tasks, but the correlation with infusion was weaker. They assert that politics and learning as compared with rational models are likely to be more appropriate for later stages of implementation.

---

36 In this respect, SISD represents an unusual hybrid of technological and organizational imperatives. The view is technologically imperative suggesting that the use of a technological technique will have a deterministic effect on successful IT development. SISD is also an organizational imperative because the promoters claim that using the tools will allow organizations to build information systems that meet their information needs.

37 MRP systems are used to plan and gather the materials required for future production runs based on orders, and a detailed inventory of the parts required to produce each product.
Chapter 2: Literature Review

Examining information/task fit, the content and format of the information produced by the system must match the task requirements: accurate, precise, current, timely, reliable, complete, understandable, and relevant. A number of IS research areas are focused on testing and improving information quality given certain users and tasks. Human/computer interaction is focused on the fit between technology's information quality and task. These studies evaluate task performance based on availability of computerized technology (system and information quality) and/or the features of two computerized technologies (information quality) (Benbasat, DeSanctis & Nault 1993). Some of the features include graphs versus text (Tan & Benbasat, 1990), color and graphics (Benbasat & Dexter 1985a & 1985b), and icons versus menus (Benbasat & Todd 1991).

Participative Design and User Involvement (PD & UI) research focuses on the interplay between the user, the developer and the information systems during design, development, and/or deployment. User involvement is a specific instance of participative decision making and group problem solving, and there is a general belief that user involvement is necessary for information systems success (Ives & Olson 1984; Baroudi, Olson & Ives 1986).

"Integrationist" and "human relations"\textsuperscript{38} approaches (Kling 1980; Hirschheim & Klein 1989) to user involvement can be considered an organizational imperative. These approaches to development constitute a bottom-up organizational imperative involving users in building information systems that meet organizational needs. Three explanations of the reasons for

\textsuperscript{38} Integrationist and human relations approaches to development realize that organizational life is interpreted by the members, and that integration can be achieved through discussion and consensus.
successful development of IS and IT with user involvement include self control (Baronas & Louis 1988), learning between both users and developers (Franz & Robey 1984; Newman & Noble 1990), and conflict resolution and accommodation (Franz & Robey 1984; Robey, Farrow, & Franz 1989; Checkland & Scholes 1990; Newman & Noble 1990). All three approaches implicitly or explicitly assume that individuals may have different ideas about successful IT implementation and the goals of the organization, and that involvement achieves consensus through discussion (Robey, Farrow & Franz 1989; Newman & Noble 1990). Variance research by Baroudi, Olson, & Ives (1986) confirms that user involvement leads to both increased system usage and information satisfaction, and that increased information satisfaction leads to increased system usage.

Therefore, like the technological imperative, the organizational imperative assumes that computing technology delivers specific impacts. Unlike the technological imperative, technology is chosen and shaped by the strategic direction of the organization. Organizational imperative processes include strategic IS planning, system analysis and design using top-down approaches, and integrative and human relation forms of user involvement.

2.4.5. Reflections on Unidirectional Theories of Technology-Context Interaction

There is widespread evidence to suggest that both the technological and organizational imperative are unable to explain aspects of ISD and computing development. The strongest evidence against the technological imperative is the high divergence of technology from

---

39 Thus, demonstrating that users are using a theory of reasoned action approach to system use.
perceived goals, including failure (Lyytinen & Hirschheim 1987). Many researchers have alluded to the failure of information technology implementation (Dutton 1981; Lyytinen 1987, Hirschheim and Newman 1991; Myers 1994a; Keil 1995). Forester & Morrison (1990) report that 75% of information systems are either not developed or if developed, are never used. Other authors report similar results (Gibbs 1994; Landauer 1995). Whether this "failure" is defined by subjective expectations, differences between stakeholder goals, or technical failure, nevertheless weakens the direct connection between context and technology described by the organizational and technological imperatives.

Second, there is a lack of consistent findings using similar techniques and technologies, implying that context may have a very important and complex effect on outcome (Orlikowski 1991). Third, an examination of a technology's origins before becoming an object, to be adopted and used, is missing. Fourth, developers' ideas and vision appear to be crucial in technology development (Schon 1983; Turner 1987; Beath & Orlikowski 1994; Petroski 1994; Thomas 1994).

Discounting of the technological imperative, however, does not imply acceptance of an organizational imperative (Thomas 1994). First, the organizational imperative cannot escape the problem of technological failure and divergence, without resorting to a call for greater use of its already widely adopted approaches. Second, system planning is missing from many successful IS and IT developments (Turner 1987; Lyytinen 1987). Some researchers remark that the belief in an organization's control of the market through the introduction of IT is very unlikely in the best case, and can be damaging to the organization in the worst case (Harrington 1991). Third, many of the system design and analysis techniques have been
criticized for their limited focus on technical issues, and ignorance of social aspects (Kling 1980; Kling & Scacchi 1982; Lyytinen 1987). In addition, system design does not generally follow the top-down and “life-cycle” approach advocated by these methods, and when it does, failure is still high (Gladden 1982; Turner 1987). There is also little evidence to suggest that these formalized techniques have significantly improved system success, and a textual analysis of these methods has revealed contradictions (Beath & Orlikowski 1994). Fourth, even though the task/technology fit literature recognizes the importance of context, its method of placing context in boxes ignores its complex interaction and influences on ISD, rendering the categories irrelevant for specific circumstances. And finally, there has been inconsistency in results of user involvement on project success if seen as a “factor” necessary for system success (Ives and Olson 1984; Brooks 1987; Hirschheim & Newman 1991).

As a result, several researchers criticize the technological and organizational imperatives as naive. Burton (1992) states:

A moment’s thought will show that both approaches [technological imperative and organizational imperative] are open to criticism. Neither, for example, pays attention to the complex inter-linking of society and technology, in which cause and effect are so difficult to disentangle, while technological determinism removes any human values from the process of technological development. (p. 22)

---

40 The life-cycle approach argues that IS development should follow three structured steps: requirements analysis, design, and implementation.

41 Checkland & Scholes (1990) comment that in the real life of managers, it is the details that matter, not generalized theories.

42 Some have criticized the lack of understanding of the different approaches to user involvement and its effect on system development success (Bostrom & Heiman 1977, Boland 1978; Olson & Ives 1984; Wynn 1991). These differences include the level, stage, quality, and intent (cooperation versus co-optation) of user involvement. Added to this would also be the characteristics of the user group, their position in the organization, and their relationship with the development team (Markus 1981). Olson & Ives (1984) also criticize the research for lacking a theoretical basis to explore user involvement and the causal mechanisms that lead from involvement to technological design and implementation.
Chapter 2: Literature Review

Thomas (1994) provides a critical comment on both approaches:

But both perspectives share a common flaw; because they focus attention primarily on the structural outcomes or impacts of change, they tend to portray the relationship between technical and social systems as static and unidirectional. By neglecting the process of change --- especially the process through which choices of technology are made—they generally fail to capture the dynamic and interactive nature of the relationship between the technical and social systems of an organization....to explain what new technology does to organizations—or as it is commonly put, ‘how technology impacts organizations’—we must also explain what people are trying to do to organizations and, by extension, to themselves by means of new technology. First, it is not enough to claim that technology ‘impacts’ organizations; it is essential to ask as well how and why particular technologies are chosen. Second, it is not enough to claim that technology is the simple product of social or strategic choice; it is essential to ask as well how technological alternatives were themselves framed, how the objectives or interests of different organizational actors shape the range of possibilities considered, and, most important, how differences in objectives or interests influence the outcomes of change. (p 4)

The technological imperative and organizational imperative fit the description of discrete-entity\textsuperscript{43} models provided by Kling & Scacchi (1982). They observe that most information systems are developed by multiple groups and thrive in multiple settings, and therefore do not meet the assumptions of the formal-rational perspective implicitly advocated by the technological and organizational imperatives.\textsuperscript{44}

The following sections explore alternative theories of technology-context.\textsuperscript{45} The alternative process theories are the emergent perspective and the social technology theory.

\textsuperscript{43} Discrete-entity models examine information technology in isolation from the context that surrounds its usage and operation.

\textsuperscript{44} Hirschheim & Klein (1989) develop four paradigms of system development based on ontology (objective/subjective) and conflict (conflict/integration). The TI and OI fit into their functionalist category where the view of the developer is that the organizational mission and goals are integrated and that the ontology is objective.

\textsuperscript{45} These process theories would be explained best by Kling & Scacchi’s (1982) web models and Hirschheim & Klein’s (1989) radical structuralist, and neo-humanist perspectives.
2.4.6. Emergent Perspective

The *emergent perspective* (EP) theory (Markus & Robey 1988) states that a complex interaction occurs between context and technology over time. In this sense, the action frontier has weakly defined boundaries, allowing a large set of task, organization, and technology configurations to emerge from this interaction. In its stronger form, EP states that technology and context are inseparable. Every context and technological appropriation and adaptation is unique. Developers and users determine the method of selection and criteria, the choice, the "choice within the choice" (Thomas 1994), and implementation approaches. The following figure illustrates the emergent perspective theory:

**Figure 2-4: The Emergent Perspective on Technology and Context**

---

*Thomas’ (1994) phrase for the second level choices made after a software platform is chosen.*
In the emergent theory, the interaction between technology and context appears somewhat or largely unpredictable. A key part of this complex interaction is that designers and users play crucial roles in design and use (Olson 1981; Nadler 1982; Franz & Robey 1984; Olson & Ives 1984; Petroski 1994; Thomas 1994).

2.4.7. Emergent Perspective Research

The research addressing emergence and the issues and methods of designing in emergent settings includes the following: the influence of the designer's world view; alternative views of system development; the predicament of design; the emergent evolution of technology; and the innovation process. Although the research is generally descriptive, prescriptive messages imply that the inability of designers to understand and deal with emergence leads to irrelevant systems that are rejected.

An increasing body of literature reveals that the designer's world view affects ISD. Given that many designers are being trained with a technical focus to development, assumptions about social, organizational, and political environments surrounding development and implementation are immature and undeveloped (Bostrom & Heinen 1977; Kling 1980; Dutton 1981; Kling & Scacchi 1982; Gladden 1982; Markus 1983; Robey & Markus 1984; Hirschheim & Klein 1989; Hirschheim & Newman 1991; Myers 1994b). Specifically, there is growing criticism over the mechanistic and deterministic view of organizations by developers. The design approach of "scientism" and technology focus is criticized by

---

47 Bostrom & Heinan (1977) are quick to suggest that the blame does not lie entirely with the developers. Complex social and environmental demands and assumptions produce a technical focus.

48 Scientism is the general approach to inquiry that focuses on universal laws of human behavior, disregarding the setting, and ignoring the human "free-will" and contextual influence on the object of study.
Bostrom & Heinen (1977), Markus (1981), Gladden (1982), Nadler (1982), Schon (1983), Klein & Lyytinen (1985), Boland (1987), Hirschheim & Klein (1989), and Harrington (1991). The misuse of the rational approach by professional groups has also been more generally discussed in philosophy (Saul 1993; Saul 1995). Prescriptive messages from this literature claim that the inability of designers to understand and deal with emergence leads to irrelevant systems that fail because they ignore or simplify users and context (Landauer 1995). A prepared developer would be able to deal with opportunities and/or pitfalls that arise during emergence.

Some researchers in the ISD and technology literature provide alternative views of design. As a result of the increased complexity of design environments and changing expectations of the methods and goals of ISD (Klein & Hirschheim 1985), a changing mindset has appeared to deal with a number of “failures” in the planning and design (P&D) fields (Nadler 1982). These include: arranging for continual change, using information from many sources, involving people, specifying and presenting a solution (communication), and focusing on solution quality, solution implementation, and effective use of P&D resources.

Brook’s (1987) suggestions for improving software design are not technological but social and context related. First, organizations should buy instead of build software, because the myriad of general purpose software packages on the market makes building less attractive. Second, iterative development and rapid prototyping should be used, because the hardest part

---

49 Harrington (1991, p. 199) suggests that designers want to implement first and ask questions later. This approach is encouraged by suppliers and software vendors.

50 Harrington (1991) provides a gloomier picture, claiming that the complex web of computing and its effect on organization is somewhat akin to a lottery, even with prepared developers and suppliers. Luck has an important part to play.
of software development is not building, but determining what to build. Third, incremental
development should be employed to “grow” software instead of build software. Finally, great
designers address an important creative process required to produce excellent software.

Hirschheim & Klein (1989) remark that most developers assume organizations are
integrated enterprises and reality is fixed and stable, a “functional”\textsuperscript{51} perspective. Alternative
views of development acknowledge the political and fragmented nature of some organizations. Some methods of applying the alternative views are through the use of “integrative”\textsuperscript{52},
“radical structuralist” and “neo-humanist” approaches.

Kling (1980), Kling & Scacchi (1982), Kling (1987), Myers (1994b) and Thomas
(1994) report that the environment around information systems development and operation is
conceived too narrowly and incorrectly. The \textit{a priori} boundaries, set by developers before
development begins, often ignore important relationships that are crucial to short-term and
long-term survival of an information system.

Kling & Scacchi (1982) compare “web”\textsuperscript{53} versus “discrete” models, and claim that
web models more directly challenge some of the assumptions of discrete entity models, and
explicitly examine those areas ignored by discrete entity models. These areas include “social
leverage”, narrow boundaries, and historical constraints and commitments. “Social leverage”
means that some groups will win while others may lose on the larger social terrain. Narrow
boundaries often fail to include suppliers and vendors and other groups essential for system

\textsuperscript{51} The “functional” perspective assumes organizations are united entities, with little conflict, and integrated goals that achieve specific ends for the organization and society. The term “functional” is most likely a short-hand for Talcott Parson’s “structural functionalism” found in the sociological literature.

\textsuperscript{52} “Integrative” implies that the organization is integrated in its views, but the views are subjective and can be altered.

\textsuperscript{53} Web models focus on the many contextual factors that develop, influence, support and sustain IT and IS within a setting.
success. Historical constraints and commitments requires an understanding of the organization's history and the emerging social relations and connections between organizational groups and with groups external to the organization. Discrete entity models either ignore or believe these relationships are forged over a short period of time. The result is a weakened link between technology and organizational structure, given the overriding importance of specific "situations" within each case:

Boland (1987) and Wurman (1990) observe that decision making is not a mechanistic mixture of data, a human receiver, and behavior.

We are fostering an image of the world in which the human meaning of knowledge and action are unproblematic, predefined, and prepackaged. (Boland 1987, p. 365).

Boland's (1987) use of the word "in-formation" implies the more idiosyncratic and direct involvement of individuals in forming attitudes, intentions, and actions using the information. "Words are symbols and meanings are always multiple and ambiguous" (p 366). The result is to place the human actor back into the information, decision, and behavior chain. Boland (1985 & 1990) provides hermeneutics\textsuperscript{54} and phenomenology\textsuperscript{55} as alternative approaches to system development.

\textsuperscript{54} Hermeneutics is the study of textual interpretation, and the process of understanding that text. Boland (1990) suggest that hermeneutics is a good method to examine the output of an information system; because it does not assume information is a reflection of reality, but needing to be constructed by the reader. Boland pulls in three views of hermeneutics: Pepper's \textit{world view} as a background metaphor that channels interpretation, Rorty's \textit{contingency} approach to language as being ever-changing and not simply representing finer grains of reality, and Ricoeur's \textit{distanciation} where the ideas and thoughts of the writer are separated in text, and readers must come to appropriate the alien text in their own way, "to make ones own what was initially alien".

\textsuperscript{55} Boland (1985) discusses phenomenology. Phenomena is the essence of our experience: that which remains after the accidents, contingencies, and pre-suppositions we bring to our every day experience. Essences are not grasped empirically, but are grasped through intuition. This cycle of essence, from experience to purified understanding, is continual. It tries to grasp the hidden assumptions behind interpretation. Tied to hermeneutics (the reading of text), it is grounded in description. It is \textit{critical} because it attempts to understand hidden assumptions about our beliefs of reality. Applied to IS, it tries to use phenomenology to understand text. Because of the historical understanding of text and information, the central problem of meaning ignored in traditional IS design is brought to the forefront with
Examining user involvement (UI), the organizational imperative approach to UI would use it as a method of building an integrated consensus of task and technology, and/or increasing system acceptance (Bostron & Heinan 1977). The research on user involvement, however, points to inconsistent findings when UI is viewed as a factor (Olson & Ives 1984; Robey & Markus 1984; Robey & Franz 1989). In addition, Hirschheim & Newsman (1991) observe that user resistance may indicate a legitimate problem with the software, integrating information systems may undermine existing commitments and power structures in the organization, the gulf between developers and users is widened by assuming the system developer knows best, IS development and politics are intertwined, and political support is necessary for any successful implementation. As a result, a more emergent view of user involvement as a complex process of information system development is mentioned in the literature. This includes “design-in-use”, “participative design”, “soft systems development”\textsuperscript{56} and “spiral development” (Markus 1981; Markus 1983; Franz & Robey 1984; Olson & Ives 1984; Boehm 1988; Gould 1988; Hirschheim & Klein 1989; Checkland & Scholes 1990; Newman & Noble 1990; Henderson & Kyng 1991; Wynn 1991; Newman & Robey 1992; Landauer 1995).\textsuperscript{57} These views of user involvement move closer to an accommodation, and “neo-humanist” and political approaches to UI. Each assumes that organizational reality is

\textsuperscript{56} A development approach focused on the human side of computing; hence the word “soft”.

\textsuperscript{57} The result of this process may be what Weiser (1991) calls *ubiquitous computing*; a vision that good computing technology disappears into the background of work because it is well designed for the task. Key to this is a match between the physical and interface characteristics of the technology and its match with the tasks and user expectations and learning.
not fixed and that organizational groups and individuals are divided in their goals and views of organizational life (Franz & Robey 1984; Hirschheim & Klein 1989).

"Design-in-use" assumes that individuals in an organization carry special knowledge of their work, including both "doing" and "understanding" (Henderson & Kyng 1991). As a result, the top-down approach to software design is unable to handle emergence: changes in situations may make the software irrelevant, complexity of the organization and environment makes it difficult to anticipate every aspect of the organization and system usage, and many different situations of use create a problem of logically deducing every aspect of the software before implementation. The "adaptation" principle of software design is important for building successful software that can handle task, user, and organizational emergence. Four characteristics of adaptable software are required: flexible (objects of the system can be interpreted in many ways); "parameterized" (providing a menu of behaviors from which to choose); "integratable" (system can be integrated into other systems/software); and "tailorable" (user can change the parts through accelerators, specialization, and added functionality) (Mohan, Holstein & Adams 1990; Henderson & Kyng 1991).

In addition to the designer's view and role in ISD, the complex environments of technology causes inherent predicaments with planning and design, implying a risk of technological divergence from original plans and failure for unprepared designers. Nadler

---

58 This is especially true for "canned" or "shrink-wrapped" software that will be used by many users in many different locations.

59 Landauer (1995) disagrees, suggesting that tailorable software can produce too many features and complexity in the software. The "tailorability" idea also assumes that users can measure and evaluate their productivity using different approaches. Human computer interaction suggests that there are universal interface designs that help everyone.

60 Henderson & Kyng mention that "object oriented" design and programming is providing many tools for addressing tailorability through inheritance, reusability, and alteration of existing objects.

The issues facing the P&D field today are very complex, with less definable problems being addressed than in the past (Nadler 1982; Schon 1983). In addition, the environments or causal texture of environment⁶¹ around technology have been categorized as more turbulent, less predictable and more inter-connected than in the past (Emery & Trist 1971; Terreberry 1971). The result has been a gross increase in the area of relevant uncertainty (Emery & Trist 1971). Combined with the complex problems is a continued focus on rational/mathematical models and theories. As a result, the following problems have appeared: concentration on solution finding; theories that fail to incorporate psychological and sociological realities of the business or professional world; classification of problems by techniques rather than ends; solutions that lack creativity; splitting of problems in specialized and piecemeal fashions; focus on the “expert” role to the detriment of other more important roles (facilitator, negotiator, coordinator, advocate, and motivator); creation of data overload; and poorly related P&D groups that require closer coordination. In addition, computing technology can radically alter the perception of individuals, producing an imbalance with the other social dimensions of the organization (Harrington 1991; Thomas 1994). Designers appear to disregard and/or downplay these dimensions (Bostron & Heinan 1977; Schon 1983; Boland 1987).

---

⁶¹ A term used by Emery & Trist to describe the complexity and independence of environmental influences around an organization that operates within that environment.
Information systems is a P&D activity with its own unique problems: abstraction and “invisibility”; correspondence with unspecified and complex human behavior; changeability; pressures to adapt and incorporate many political views; dependence on a rapidly changing software environment;\(^62\) and correctness subject to verification (Brooks 1987; Turner 1987; Forester & Morrison 1990\(^63\); Harrington 1991; Landauer 1991; Gibbs 1994; Keil 1995).\(^64\)

Central to many of these issues is that system development involves both problem definition and problem solution simultaneously (Turner 1987; Thomas 1994). Candidate ideas are presented, creative leaps and prior experience are employed to reach these candidate ideas, and prototyping is used to test these ideas (Landauer 1991). Therefore, the predicament of design produces emergence.\(^65\)

Also supporting emergence is the literature examining the evolutionary and emergent development path of technologies. Several researchers discuss the history of technological evolution from an emergent perspective (Kidder 1981; Latour 1987; Harrington 1991,

---

\(^62\) Dutton’s work (1981) provides some interesting insights into the implementation failure of a land planning system (Growth Guidance System - GCS) in the Tulsa municipal government. Dutton’s case suggests that the shifting context surrounding IS and IT affects the “appropriateness” of the technology. The CGS was originally designed in a municipal context of controlled land development. When a new pro-growth mayor was elected close to the end of the CGS development cycle, the software was reused to support the pro-growth approach, illustrating the malleable nature of the CGS. Without direct top-level support, however, the system slowly floundered and was abandoned. This case illustrates an emergent perspective of IS.

\(^63\) Forester & Morrison (1990) state that software is prone to total rather than partial failure, and enormous complexity means that software can never be fully tested before use.

\(^64\) Landauer (1991) suggests that general theory in human-computer interaction is impossible, and that homely and minor impact theories have proven more useful. Many interface ideas emerge from lucky hunches, and iterative testing and analyzing with users.

\(^65\) Newman & Robey (1992) state that a complex process of development does not imply chaos and unpredictability. Each stage of development leads to a certain probability of other stage appearing. Their research examining the stages of development in two cases was somewhat predictable from theory. The nature of environmental shocks and the comparison of initial and final plans, however, produces emergence.
Franklin 1992; Latour 1994, Petroski 1994; Thomas 1994; Keil 1995). This emergence results from the influence of many groups and individuals involved in design and usage of the technology over time.

Kidder (1981) describes the history and people involved in building a Data General computer, and the impact of events, conflict, and controversies on design. Latour (1987 & 1994) describes the controversy surrounding technology and science, and the disappearance of these controversial decisions into the technology long after the controversy has disappeared. Basalla (1988) and Petroski (1994) examine the diversity and evolution of various technologies including the paper clip, stapler, barbed wire, and the cotton gin. Both conclude that technology emerged from previous natural and man-made templates and evolved through usage and modification by many individuals. Both authors consistently demonstrate how form does not follow from function, as demonstrated by the diversity of tools used for the same task. This emergence of technology for a task is the norm in technology development. Franklin (1992) provides a critical examination of technological evolution by examining environmental and social conditions that have affected how we perceive, behave, and modify technology. She examines the evolution from ‘holistic’ to

---

66 Keil’s “escalating commitment” of software development provides a theory of “run away” ISD.

67 This research can be linked with the concept of *ubiquitous computing*, where good technology becomes hidden and embedded in organizations and society (Winograd & Flores 1987; Weiser 1991).

68 Petroski hesitates to use the word ‘evolution’, because to him, it implies random variation. Instead, he states that variation in technology is driven by annoyance, market differentiation, and purposeful action.

69 Petroski recounts the story of a designer called to the witness stand in a client’s case against a manufacturer who had stolen his client’s design. The defense’s case rested on the inability to produce any other design that would be practical and functional. When the designer was called to the witness stand and asked if he could draw alternative designs, the designer recounted: “I unfolded my easel, placed the drawing board on it, and started making rapid sketches in large black outline, visible to anyone in the back row. Ten minutes later, I had about twenty-five designs, all different, most of them attractive, all of them practical” (p. 178)
'prescriptive' technologies as being historically based on the division of labor, the industrial revolution, and the power struggle between capitalists and workers. The result of unguided technological development has produced such unplanned and undesirable properties as pollution, fragmented communication, and poor coordination.

Several IS researchers have alluded to IT/IS as an innovation (Kwon & Zmud 1987; King, Gurbaxani, Kraemer, McFarlan, Raman, & Yap 1994), implying that design and the interaction between technology and context is emergent. The term “innovation” is defined by Van De Ven (1986):

the development and implementation of new ideas by people who over time engage in transactions with others within an institutional order (p. 590).

Van De Ven (1986) discusses several key problems in the process of managing an innovation, providing useful insights on the developer’s role linking context to technology. First, the problem of “managing attention” is based on the individual’s physiological limitations to handle complexity. Key to the human attention problem is the slow reaction of groups to deteriorating situations, and the crisis approach to innovation that emerges when a full-blown catastrophe strikes. Second, while idea generation is an individual activity, converting ideas into “good currency” requires commitment and collective agreement amongst many groups and individuals. This requires an understanding of the stakeholders around an innovation and the process of involvement bringing these groups together in coalitions. Third, management of part-whole relationships warns that the project should not lose focus of its larger purpose and direction. More often than not, large projects are split into separate tasks to be handled by various groups, with the result that the integration of the parts produces poor
results. It must be understood that most projects are a collective achievement, and require the understanding, commitment, and collective wisdom of many groups. Finally, institutional leadership and an innovation context are required to allow the innovation to flourish. Mission, purpose, values, and excitement are required in the organization to create an environment that supports innovation, and to foster innovations that support the environment.

Contextual events and structures around a project can substantially influence the learning of a development team. Van De Ven & Polley (1992) longitudinally examined the learning that occurs around a new innovation over an eight-year period, examining the contextual structure surrounding the project leading to poor learning and product failure. The macro-structure around the project inhibited early learning about the innovation’s problems, as the innovation team continued to ‘sugar-coat’ the results, and activities concentrated on maintaining funding. In addition, the noisy nature of feedback confused the project leaders as good, bad and mixed signals poured into the meetings.

Learning an innovation’s capabilities regarding work is important to technological adjustment. Attewell (1992) focuses on the supply side factors and its effect on IT development, adoption and infusion. He calls his approach the knowledge-barrier institutional-network approach. In this theory, communication is viewed differently than the “signaling” and “knowledge transfer” approaches in classical diffusion theory. Learning how a new technology can improve organizational productivity involves a slow process of

---

70 A key concept in Van De Ven’s management of the part-whole relationship is the “hologram” (Morgan 1986); where key elements of the entire project are present with the subunits working on the tasks. The linear approach to project development, from research to development to marketing, should be replaced by simultaneous coupling. Redundant functions allow key elements to be socialized into group members. Requisite variety requires the complexity of the real world to be built into the unit.
organizational learning called "learning by doing" (Carrol & Mack 1984). Frequently, unexpected changes in organizational structure and processes are difficult to predict and this change must be learned through a painful readjustment. The result is that technology can be either "competence-enhancing" or "competence-destroying", building on previous work methods or rendering previous work methods obsolete. In addition, Attewell (1992) identifies that technology needs to be learned in the setting, to determine its capabilities and interaction possibilities with other systems, a process called "learning by using". The result is that a large amount of knowledge is not transferable, but specific to the organization.

Attewell proposes the following consequences of the knowledge-barrier institutional network approach: (1) technical knowledge needs to be relearned and reinvented by the organization; (2) the burden of organizational learning becomes a barrier to adoption and adaptation; and (3) the supplier must provide more than just the equipment, software, and some information about benefits and technical usage.

To summarize, the emergent perspective differs from the technological imperative and the organizational imperative by suggesting that context and technology interact in a complex and somewhat predictable process. Prescriptive messages from this literature remark that the inability of designers to understand and deal with emergent shocks from the environment and learning will most likely lead to irrelevant systems that are rejected. An emergent environment around technology may produce opportunities and/or pitfalls that a prepared developer may be able to deal with effectively.

2.4.8. Social Technology

DeSanctis & Poole (1994) construct a fourth theory, the social technology (ST)
theory, which posits a "soft-line determinism" between technology and context. Equal focus is provided to both the constraining and enabling features of a technology and context. This fourth view of the technology/context link is diagrammed below:

**Figure 2-5: Social Technology Theory**

![Diagram](image)

In the ST theory, technology provides a larger domain of action than the technological or organization imperatives, but still has a definite boundary. Actions closer to the boundary are less likely to occur, and actions outside the boundary are impossible. Surprises emerge when action appears in an undesirable position within the action frontier.
2.4.9. Social Technology Research

Social Technology studies of IS/IT development tend to arise from descriptive accounts of technological implementation and technology's effect on or reaction from organizations. The interaction between technology and context in these studies resembles emergence, but from identifiable characteristics such as mismatches between organizational structure and technology. The research on social technology comes from two areas of research: adaptive structuration and structural power. Both suggest that the interaction between structure and technology is explainable, but only after implementation has occurred.\(^71\)


One of the earlier discussions on the constraining and enabling role of technology and context was Thompson's (1967) work. According to Thompson, the main goals of the rational organization are to reduce the uncertainty around the cause-effect relationship between technology and production (technical rationality), while maintaining a product/service that was relevant to the supplier and consumer environment (organizational rationality). Classifying technology into three types, long-linked, mediating, or intensive

---

\(^71\) The research assumes that enough empirical evidence will determine the capabilities of a technology in certain organizational settings, and/or the transmission mechanisms between technology and context that can be eventually controlled by developers (Orlikowski & Robey 1991; Orlikowski 1992a; Orlikowski 1992b).
technologies\textsuperscript{72}, Thompson provides various propositions about the proposed fit between core technology, organizational structure and environmental conditions, providing some early ideas of adaptive structuration.

Moving specifically to the structuration of IT and context, Orlikowski’s (1992b) study of Lotus Notes in a consulting organization points out the unexpected but understandable effect of structure on groupware technology adaptation. Because the culture of competitiveness provided few incentives for sharing information with Lotus notes, the software failed to be adapted and infused into the organization. She comments:

where the premises underlying groupware are incongruent with those of the organization’s culture, policies, and reward systems, it is unlikely that effective cooperative computing will result without a change in structural properties (p. 368).

Tyre & Orlikowski (1994) provide a general description of the interaction between technology and organizational structure through the stages of adaptation and routinization. Challenging the view of continuous and rational acceptance and adaptation to new technology, they studied three organizations: one implementing new precision metal tools, one implementing CASE\textsuperscript{73} tools, and one implementing personal computers with users at a university. Their findings demonstrate that short and dramatic changes in work occur at discontinuous points over time. They conclude that the following factors influence this unexpected pattern of adaptation: the need and drive for production impedes adaptation; unchecked work patterns congeal and constrain further adjustments; expectations adjust

\textsuperscript{72} Intensive technologies are the best definition of the core technology used in hospitals.

\textsuperscript{73} Computer Aided Software Engineering that support analysis and design.
downward to match less than optimal work patterns; and team membership and enthusiasm erode quickly.

Zuboff (1988), in her examination of IT implementation in several companies,\textsuperscript{74} discusses the potential for IT either to automate or to “informate” work.\textsuperscript{75} Automation causes the displacement of human labor, and is not unique to IT. “Informating” is a unique characteristic of IT, where the technology produces information about the process and itself. She concludes that IT has the potential to transform the nature of work in the above two directions (automation or informating) based on the implementation and usage of the technology by organizations.

Barley (1986) discusses the varying effects of CT scanner technology on the organizational structures and the process of change at two hospitals. Using longitudinal research methods, he shows how in each case, organizational structure affected and was affected by the characteristics of the staff and previous work patterns.

Structural power research suggests that IS can influence the balance of power within an organization, thus contributing to resistance and rejection of new IS and IT. Markus' (1981) historical and longitudinal analysis of a Material Requirements Planning (MRP) system in two manufacturing plants within a single company reveals power and political reasons for user resistance and acceptance. The new centralized MRP system violated Plant One’s power as the lead plant in the production process feeding Plant Two, while Plant Two accepted the...

\textsuperscript{74} Pulp and paper, telecommunications, dental claims, stock and bond transfer, bank, and pharmaceutical companies,

\textsuperscript{75} Harrington (1991) also discusses the ability of IT to produce broader awareness and perception of environment, versus more general technology. Both Zuboff and Harrington implicitly state a moderate technological imperative with emergent properties.
system readily because the centralized information allowed it to monitor Plant One’s activities. Markus (1983) found similar results in the development of a cost accounting system.

Thomas (1994) proposes a process-power perspective where power lies at the root of the manufacturing innovations he studied. Usually engineers and developers of technology lead the innovation to enhance their position in organizations. Responses from other groups to protect and/or enhance their position by shaping the technology’s further development and adoption describe the ensuing development path.

The social technology perspective takes an intermediate stance between the emergent perspective and the organizational imperative. Technology constrains and enables action in planned and unplanned ways. Adaptive structuration research falls within the social technology perspective. The interplay between technology and organizational structure produces planned and unplanned outcomes. The structural power literature suggests that systems pose opportunities for power posturing within an organization, and that the resistance to or acceptance of a technological system can be explained by referring to a group’s power and aspirations.

2.4.10. Conclusion: Reconciling the Four Theories of Technology and Context Interaction

Reconciling these four views of the technology and context is an important goal of this thesis. The result of such an analysis may be one of the following:

---

Markus suggests that three areas of power are influenced by IS: decision making, behavior and performance, and symbolic power.
Table 2-5: Reconciling the Four Theories

<table>
<thead>
<tr>
<th>Reconciliation Method</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutually Exclusive</td>
<td>Theory OI explains the technology-context process better than theory TI, EP, or ST.</td>
</tr>
<tr>
<td>Equal Validity</td>
<td>Theory ST and theory OI explain different aspects of the process.</td>
</tr>
<tr>
<td>Special Case</td>
<td>Theory ST is a special case of Theory EP.</td>
</tr>
<tr>
<td>Association</td>
<td>Theory ST occurs before Theory OI over time.</td>
</tr>
</tbody>
</table>

The purpose of this thesis is to reconcile these four theories and the various IS literature, addressing IS development by comparing and contrasting the data from two case studies. Through this method, I hope to understand and integrate the current IS literature on system development and provide a conceptually rich explanation of the development process.

2.5. Chapter Summary

Addressing the first motivation of producing a conceptual language and framework of the ISD research and theory, this chapter described an implementation framework, two descriptive constructs to simplify technology and context (action and "action frontier"), four theories linking technology and context (technological imperative, organizational imperative, emergent perspective, and social technology), and information systems research that employs these theories. The result is a framework, map, and a language for discussing the case studies.

The following table provides a summary of the research areas under the four theories:
Chapter 2: Literature Review

Table 2-6: Summary of Four Process Theories and ISD Research

<table>
<thead>
<tr>
<th>Process Theory</th>
<th>Research Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technological Imperative</td>
<td>Technology</td>
</tr>
<tr>
<td></td>
<td>Systems</td>
</tr>
<tr>
<td>Organization Imperative</td>
<td>Strategic Information Systems Planning</td>
</tr>
<tr>
<td></td>
<td>Structured Design and Analysis</td>
</tr>
<tr>
<td></td>
<td>Task-Technology Fit</td>
</tr>
<tr>
<td></td>
<td>Integrative and Functionalist Forms of User Involvement</td>
</tr>
<tr>
<td>Problems with Unidirectional Theories</td>
<td>Technology Divergence</td>
</tr>
<tr>
<td></td>
<td>Evidence Against top-down Planning</td>
</tr>
<tr>
<td></td>
<td>Conceptual Problems</td>
</tr>
<tr>
<td>Emergent Perspective</td>
<td>Designer’s View</td>
</tr>
<tr>
<td></td>
<td>Alternative Views</td>
</tr>
<tr>
<td></td>
<td>Predicament of Design</td>
</tr>
<tr>
<td></td>
<td>Evolution of Technology</td>
</tr>
<tr>
<td></td>
<td>Political forms of User Involvement</td>
</tr>
<tr>
<td></td>
<td>Innovation</td>
</tr>
<tr>
<td>Social Technology</td>
<td>Adaptive Structuration</td>
</tr>
<tr>
<td></td>
<td>Structural Power</td>
</tr>
</tbody>
</table>

The final section provided possible methods of comparing and integrating the various process theories that will be explored in the next chapter.
3. **Research Question and Methodology**

3.1. **Research Question**

The following research question is addressed in this thesis:

(1) What is the nature of the interaction between technology and context during ISD?

The purpose is to generate new hypotheses about the technology-context interaction for future research.

3.2. **Methodology**

To address this question, two cases of software development were studied and are described using a *comparative case*, *action research*, and *longitudinal analysis* approach. *Data collection* and *data analysis* strategies are also discussed.

3.2.1. **Comparative Case Design**

The purpose of this research is to explore the interaction between technology and context across time (Franz & Robey 1987). As a result, this thesis uses a multiple case study approach to compare and contrast the researcher's development experiences in two software development cases (Yin 1994). A case study approach is considered an appropriate method for studying a phenomenon in its natural setting, for employing multiple data collection techniques, uncovering the complexity and richness of a phenomenon, and exploring and generating hypotheses (Benbasat, Goldstein, & Mead 1987; Yin 1994). The purpose of the two case studies is not to produce two isolated stories, but to link them to the theory and
empirical work in the IS field through “analytic generalization” and other methods (Markus & Robey 1988; Eisenhardt 1989; Lee 1989; Yin 1994; Sabherwal & Robey 1995). These methods include the search for procedural patterns, the use of rich description to illustrate the data, the use of theory and empirical work to help explain and provide links for generalization, and the comparison of two case studies to allow replication.

Case Study One (SoftHeart) was chosen based on accessibility and an opportunity to explore a potential ISD project within a clinical setting. The approach for Case One is the “information gathering” approach (Layder 1993) and what Yin (1994) calls an exploratory case, where data are collected using few preconceived notions in a specific problem area to produce new theory (Eisenhardt 1989). In this thesis, the problem area is software development and early implementation. Using the “information-gathering” method, a grounded theory approach is used for Case One, using theory to help interpret and provide a language for exploring the case observations and events.

During Case One, the theoretical framework was developed, based on my literature search and review (see Chapter 2). The literature draws from a wide range of ISD and technology research to provide a rich theoretical framework, and to provide for creative juxtaposition of data and research (Eisenhardt 1989). Case Study Two was then chosen strategically to extend the emergent theory (Yin 1994). As a result, Case Two and the continued examination of Case One provide a structured examination of the theoretical

---

77 Generalization to theory and literature instead of through statistics.
78 Yin talks about analytic generalization to theory, not statistical generalization.
79 My initial approach in case one was a technological imperative.
framework. A model integrating the four process theories is then provided at the conclusion of both studies (see Chapter 6). The following diagram illustrates these three phases:

Figure 3-1: Research Design

The two cases vary on a number of theoretical and *a priori* dimensions to allow the theory to emerge from experiences in diverse settings. A summary of the case studies is provided below.
Table 3-1: Development Environment

<table>
<thead>
<tr>
<th>Dimension/Case</th>
<th>Case 1</th>
<th>Case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting of development</td>
<td>Two heart disease outpatient clinics - hospital setting.</td>
<td>Health promotion department on campus.</td>
</tr>
<tr>
<td>Setting of usage</td>
<td>Two heart clinics</td>
<td>Two Canadian communities.</td>
</tr>
<tr>
<td>Software/Technology</td>
<td>Electronic patient record (SoftHeart)</td>
<td>Health promotion planning tool (EMPOWER)</td>
</tr>
<tr>
<td>Purpose</td>
<td>Diagnosis, treatment, and patient education</td>
<td>Teach and assist planning</td>
</tr>
<tr>
<td>Task</td>
<td>Clinical, administrative, and research</td>
<td>Develop complete health promotion plans</td>
</tr>
<tr>
<td>Product</td>
<td>Tailored software to setting</td>
<td>Generic software to be diffused to many settings</td>
</tr>
<tr>
<td>IS/IT Concepts</td>
<td>Database - executive information system</td>
<td>Planning tool with decision support resources.</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>Computer network</td>
<td>Standalone</td>
</tr>
<tr>
<td>Project Team</td>
<td>LipidLink group (network infrastructure), two programmers, interviews with clinic personnel.</td>
<td>Health promotion researchers, programmer</td>
</tr>
<tr>
<td>Software Tools</td>
<td>Foxpro database, graphics server tool, Windows, and OS/2</td>
<td>Toolbook and Windows</td>
</tr>
<tr>
<td>IT/IS Starting Point</td>
<td>DOS standalone patient record in Clipper with 4 Modules (lab, patient, physician, medications)</td>
<td>Untested US version of software</td>
</tr>
<tr>
<td>Development Approach</td>
<td>Bottom Up</td>
<td>Theory and research model.</td>
</tr>
<tr>
<td>User Involvement</td>
<td>Design, development, deployment</td>
<td>Some development and deployment</td>
</tr>
<tr>
<td>My Role</td>
<td>Designer/Developer/Programmer</td>
<td>Developer and Programming Consultant</td>
</tr>
</tbody>
</table>

The level of analysis in this thesis is what Layder (1993) calls a “situated-activity” level-of-analysis, where the interaction between individuals and the outcomes of this interaction are the focus of data collection to provide an embedded case design (Yin 1994).\(^{80}\)

\(^{80}\) I use the term *events* to represent instances of situated activities, or what I call *processes*. This fits with Argyris, Putnam, & Smith’s (1985) method of dealing with the subjective nature of data by referring to the ‘ladder of inference’. Events are relatively objective statements at the lower end, and processes are interpreted and generalized event types.
The focus is on:

emergent meanings, understanding and definitions of the situation as these
affect and are affected by contexts and settings (above) and subjective
dispositions of individuals (below) (Layder 1993, p 72).

3.2.2. Action Research

This study also employs a specialized form of case research referred to as
"implementation", “problem-oriented”, or “action” research (Gibson 1975; Wolf & Rossini
1977; Susman & Evered 1978; Sandberg 1985; Wood-Harper 1985; Benbasat, Goldstein, &
Mead 1987; Galliers 1990; Jönsson 1991).\footnote{Action research (AR) is also an established method in health promotion called “participatory” research (Green, George, Daniel, Frankish, Herbert, Bowie & O’Neill 1995), with similar roots (Lewin 1945).} Action research (AR) has been defined as:

an inquiry into how human beings design and implement action in relation to
one another. Hence it is a science of practice (Argyris, Putnam, & Smith 1985,
p. 4).

The purpose of AR is both to observe and to promote change in the organization.

Action researchers point to this dual role of the research:

Action science attempts both to inform action in concrete situations and to test
general theory (Argyris, Putnam, & Smith 1985, p. 5).

On the one hand, we sought to observe, record, and interpret behavior and
events longitudinally...to develop concepts and relationships and to test
working hypotheses toward the building of a theory that would be closely
grounded in the data and that would yield immediately useful findings for
practitioners concerned with the implementation of models. On the other
hand, we were committed to assisting and facilitating the implementations. In
this respect we worked closely with and between the model builders and the
users and undertook to exert influence on the parties as it seemed appropriate
(Gibson 1975, p. 53).

The method of AR begins by analyzing ad-hoc and permanent meetings between
various participants, and intervening in one or more of these groups within a “client-system”
(Argyris, Putnam & Smith 1985). Client systems could include one group, an organization, a
network of organizations, or a community (Susman & Evered 1978). Susman & Evered
(1978) identify five cyclical phases to action research: diagnosing, action planning, action
taking, evaluating, and specifying learning.

AR then falls into four categories based on the study’s emphasis: “diagnostic” action
research focuses on the researcher collecting diagnostic data for the client and the reporting of
this data, “empirical” action research examines the actions naturally taken by the client, and
the diagnosing and reporting of this data back to the client, “participant” action research
occurs when the researcher is a part of diagnosing and carrying out a change with the client,
and “experimental” action research carries the researcher through the phases of diagnosing,
acting, and evaluating the intervention through experimental methods. Originally, this thesis
began with an experimental action research focus and switched to a participant form when
development became the focus.

The philosophical foundations of AR are praxis, critical theory, interpretive theory,
hermeneutics, pragmatism, process philosophies, and phenomenology (Susman & Evered

(1) AR focuses on what Aristotle called “praxis” knowledge, or the knowledge of action
and conduct (Susman & Evered 1978; Angeles 1981). As a result, AR is pragmatic in
assigning truth and value to a statement by its practical application to achieving new
desired ends (Jönsson 1991).

(2) AR tends toward critical theory in identifying hidden dimensions through
emancipatory knowledge, and rephrasing individual and organizational problems in
new ways to allow action (Argyris, Putnam & Smith 1985; Lyytinen & Klein 1985; Orlikowski & Baroudi 1991; Myers 1994a, 1994b). The generalization of these findings is created by drawing out the principles of intervention and action from the case settings.

(3) Findings from AR research are also potentially generalizable to at least the organizational level through an interpretive approach by searching for shared norms, values, opinions, and understandings of the organization (Sanday 1979; Lee 1991; Orlikowski 1991). In this interpretive approach, AR uses hermeneutics, suggesting that knowledge is framed by assumptions, that a holistic understanding of the context is required to understand events and specific instances, and that various individuals in a setting will disagree on meaning (Boland 1985, 1990). AR also gives priority to personal subjective experience in creating knowledge, and thus includes the principles of phenomenology. The principles of phenomenology have been applied in a number of social science settings, including the early action research of Lewin (1946).

(4) AR encompasses the “process view” where all knowledge and action is limited by its spatial and temporal context, thus suggesting that you cannot step into the same social system twice (Susman & Evered 1978).

There are a number of advantages to using an action research approach in the study of software initiation and creation.

---

82 Interpretive research is becoming more prevalent in the IS literature (Walsham 1995).
(1) AR is considered valuable for information systems designers who have been unable to utilize the more abstract IS research to deal with system design and implementation (Jönsson 1991).

(2) AR allows in-depth understanding of the consultant's role that has been poorly covered in the literature (Jönsson 1991).

(3) The goals of AR focus more on the neglected aspects of knowledge, with its focus on understanding versus prediction, making things happen versus prediction, conjectures versus deduction/induction, engagement versus detachment, and action versus contemplation (Susman & Evered 1978). As a result, action research is future-oriented, collaborative, implies the development of enabling and learning systems, generates theory grounded in action, and provides guidance and methods for tailoring development to specific situations.

(4) AR provides rich data to compare with theory by allowing researchers to observe processes and unplanned discovery (Jönsson 1991).

(5) The phenomenological and hermeneutic approach of AR has been identified as an important but overlooked method of studying and understanding information systems development and usage (Boland 1985, 1987, 1990; Myers 1994b).

(6) AR combines both observation and the facilitation of implementation to produce change in the host organization (Gibson 1975). This allows a positive interplay between theory and observation, learning, and action (Jönsson 1991).

(7) The predominance of positivistic approaches to research in information systems has
led to a philosophically restrictive view of information systems research that needs to be addressed by more research in the interpretive and critical theory arenas (Orlikowski & Baroudi 1991).

(8) Information system development has been considered an intervention into a social system where the developer acts as a change agent (Bostrom & Heiman 1977).

To limit some of the biases and weaknesses of the action research approach, this thesis employs a number of Gibson’s (1975) and Sandberg’s (1985) suggestions:

(1) assume knowledge can be extended beyond local settings and isolated stories (Schon 1983; Markus & Robey 1988).

(2) develop an action part subordinate to the goals of action.

(3) develop a theory part subordinate to the goals of scientific inquiry.

(4) create a planned interplay between the two.

(5) record processes in a manner that is useful to scientific inquiry.

(6) maintain enough professional autonomy.

Important conceptual approaches for allowing scientific inquiry to emerge from action research can be drawn from McClintock, Brannon, & Maynard-Moody (1979), Lee (1991) and Sabherwal & Robey (1995). First, McClintock et al. (1979) suggest a longitudinal case study does not need to be treated as a single instance, but the richness of the events and processes can be used to find many instances of a theme or idea. Second, Lee (1991) provides a method of integrating subjective, interpretive, and positivistic understanding of data. The subjective approach attempts to understand the individual perspective of those
involved in the setting. The interpretative approach is the researcher’s initial understanding and rephrasing of the subjective. The positivistic approach formalizes the interpretative understanding, using scientific and mathematical language. This thesis uses the subjective and interpretive approach to describe the process of ISD, providing pointers to future positivistic and variance research. Finally, future research of the qualitative data would employ Sabherwal & Robey’s (1995) method of quantifying the data within the two cases to uncover statistical patterns.

3.2.3. Longitudinal Analysis

Both cases involve the observation and collection of data continuously during the initiation, creation, and development stages. This longitudinal or “time series” design (Yin 1994) allows an examination of the software and organizational process as it unfolds, and addresses an important need for this type of research (Pettigrew 1979; Kling & Scacchi 1982; Das 1983; Pettigrew 1985; Vitalari 1985; Markus & Robey 1988; Monge 1990). Longitudinal research designs address important questions such as time, learning, adaptation, evolution, technology and social systems interaction, rates of change, and responses to contextual factors (Vitalari, 1985; Yin 1994). Various longitudinal methods are used in this study (Pettigrew 1979; Barley 1986, 1990).

3.2.4. Data Collection and Analysis

Data sources collected from both cases studies include observations of meetings, patient/provider interactions, clerical work, prototype demonstrations and usage, team meetings, documents, field notes, interviews, diaries, and design notes. This data is
triangulated to produce more valid substantive and theoretical ideas about the cases (Jick 1979; Benbasat, Goldstein, & Mead 1987; Eisenhardt 1989; Yin 1994).

Data is collected from multiple levels including individual, group, organization, and the health environment, including literature about heart disease, breast cancer, health behavior, and health promotion. This provides a rich context for a complete understanding of the context around technology and ISD (Markus & Robey 1988; Ritzer 1992; Layder 1993).

For data analysis, diary notes and summaries of documentation were typed into a word processor, ordered by time. This allowed a scanning for patterns within the data, and the recognition of critical and illustrative events and processes. In addition, an annotated bibliography was constructed to allow quick scanning of the research literature during coding. Diary notes for both cases exceeded 800 pages. Documentation of design notes, database schemas, reports, and meeting minutes exceeded 500 pages. Typewritten summaries of the diaries and documentation summaries produced 189 single-spaced pages of data for the SoftHeart case, and 57 pages for the Empower project. The annotated bibliography exceeded 50 pages.

The analysis of the data uses the 'ladder of inference' described by Argyris, Putnam & Smith (1985) and employed by Newman & Robey (1992). The following diagram illustrates the ladder of inference used in this study:

\[ \text{Diagram} \]

---

83 See Appendices C & D for a list of the health care and health promotion literature read and used in both cases.
"Process stages" are time periods where the project maintained a certain ambiance. For example, "project emergence" is a process stage in both cases. Process stages are generated from both key events and processes that altered the project’s direction. Critical and illustrative events are extracted from the cases (Miles & Huberman 1994). These events are described in the case write-ups of Appendices A and B (Eisenhardt 1989; Yin 1994). Events are then gathered into “processes” (Newman & Robey 1992). Each process is described and characterized by time, “technological levels” affected, and contextual dimensions involved.

---

84 Events are similar to Newman & Robey’s (1992) encounters, and stages matches their definition of episodes. Because these authors are examining the entire system development as a single process, and focus on the interaction between users and analysts, they determine and categorize their episodes based on a single event. These events types include: user-led, system-led, and joint-led.
Following the identification of a process, relevant literature is cited with a process, and the categorization of that literature is used to determine one or more of the four process theories that best describes the process. Using four divergent theories and two cases ensures that bias toward the data is reduced and theoretical opportunities are increased (Eisenhardt 1989; Yin 1994).

A series of inference rules allows movement up the ladder from events to processes, processes to the literature (Chapter 4 & 5), and literature to the four process theories (Chapter 2: Table 2-4). As an example of the approach, the following table outlines the three characteristics of a hypothetical process with a process called “search for software platform”:

<table>
<thead>
<tr>
<th>Question/Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contextual Dimensions and/or Contextual Level</td>
<td>Developer and software supplier x.</td>
</tr>
<tr>
<td>Technological Level and/or characteristics</td>
<td>System Functionality</td>
</tr>
<tr>
<td>Time</td>
<td>January to February</td>
</tr>
<tr>
<td>Process Theory</td>
<td>ST</td>
</tr>
</tbody>
</table>

One of the key characteristic of a process is the “levels of technology” involved in the process. The term ‘levels’ is used to describe the macro characteristics of the hardware and software infrastructure, and micro characteristics of the software program and interface influenced by the macro. Certain process types may appear with certain levels of technology while disappearing with others. The following figure illustrates the nested levels of technology, with the micro characteristics at the center, expanding to the macro
Processes are then combined to reach conclusions about each stage, and conclusions about the entire ISD process. Theoretical patterns in each case are explored by examining the pattern of the four theories across time. Theories can then be compared with "mutual exclusive", "equal validity", "association", and "special case" patterns.

Chapters 4 and 5 describe the processes, stages, and ISD process in each case based on the detailed events described in Appendices A and B. Numerical counts are used to illustrate patterns in the subjective extraction and coding of processes, providing results for hypothesis generation. Chapter 6 compares and contrasts the four process theories and their
explanation of the processes, stages and ISD process in both cases, and provides an integrated theoretical model. Percentages of the data explained by a process theory are included.

### 3.3. Some Limitations of the Methodology

Various weaknesses of the case study methodology have been documented (Sandberg 1985; Benbasat, Goldstein, & Mead 1987; Jönsson 1991; Yin 1984), including increased subjectivity, and the need for careful generalization of results beyond the cases through theory. Weaknesses of action research include its lack of control over independent variables, its interference with the object of study, its low generalizability to other settings without careful qualification, and the potential alignment of the researcher with specific interest groups (Gibson 1975; Susman & Evered 1978; Sandberg 1985; Galliers 1990; Jönsson 1991). Thus, the thesis provides ideas for hypothesis generation rather than concluding with certainty the magnitude of the four process theories in explaining the ISD trajectory.
4. Processes in the Initiation and Development of “SoftHeart”

The development of SoftHeart involved a series of “processes” between technology and context over time. Important in these processes were various stakeholders including the developer (myself), the developer’s thesis committee, clinic physicians, two programmers, clinic personnel, the LipidLink and research group, funding agencies, internal hospital departments, external government agencies, health researchers, and software suppliers.

This chapter reports on critical and illustrative technology-context processes in the development of SoftHeart. Processes, process stages and the ISD process are discussed using the four process theories and research of chapter 2. Citations and inference rules to the literature for each process are provided.

First, the chapter describes the two heart clinics, including their past, present and perceived future information systems, and the role of SoftHeart in the current and future information system. Second, the majority of the chapter examines the technology-context processes that appeared in process stages during the development of SoftHeart. The following table summarizes these process stages:

---

85 At the time of writing this thesis, SoftHeart had reached the early implementation stage.

86 Appendix A provides a more detailed account of the important chronological events in the SoftHeart case, and provides the raw data used to build this chapter.

87 Note that these stages, although presented sequentially, were often overlapping.
Table 4-1: Process Stages in the Development of SoftHeart

<table>
<thead>
<tr>
<th>Process Stage</th>
<th>Time Period</th>
<th>Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Emergence</td>
<td>April 1990 - August 1992</td>
<td>Initiation</td>
</tr>
<tr>
<td>Health Funders Skeptical</td>
<td>January 1993 - March 1993</td>
<td>Initiation</td>
</tr>
<tr>
<td>Our Response</td>
<td>April 1993 - October 1993</td>
<td>Initiation</td>
</tr>
<tr>
<td>Old and New Environmental Shocks</td>
<td>August 1993 - March 1994</td>
<td>Initiation</td>
</tr>
<tr>
<td>Decision to Move Ahead</td>
<td>April 1994 - August 1994</td>
<td>Adoption &amp; Adaptation</td>
</tr>
<tr>
<td>Prototype Presentation Sparks Interest and Constraint</td>
<td>September 1994 - October 1994</td>
<td>Adoption &amp; Adaptation</td>
</tr>
<tr>
<td>Conflict Erupts</td>
<td>October 1994 - February 1995</td>
<td>Adoption &amp; Adaptation</td>
</tr>
<tr>
<td>Decision to Act</td>
<td>February 1995 - March 1995</td>
<td></td>
</tr>
<tr>
<td>Project Builds Momentum</td>
<td>Nov. 1994 - April 1995</td>
<td></td>
</tr>
<tr>
<td>Development Quagmire</td>
<td>March 1995 - August 1995</td>
<td>Adaptation, Acceptance and Routinization</td>
</tr>
<tr>
<td>Recovery and Slow Growth to Early Implementation</td>
<td>August 1995 - Present</td>
<td>Routinization and Infusion</td>
</tr>
<tr>
<td>What Lies Ahead</td>
<td>Present</td>
<td></td>
</tr>
</tbody>
</table>

Finally, the chapter concludes with a discussion of the four process theories’ description and illustration of the SoftHeart case.

4.1. The Two Heart Clinics and SoftHeart

There are two heart clinics involved in the development and usage of SoftHeart. Clinic One has been in the project since the summer of 1992 when the developer first visited its site at Hospital One. When Hospital One was closed in the spring of 1994, Clinic One was moved to Hospital Two, and located beside Clinic Two. In March of 1995, after talks with the director and staff of Clinic Two, the SoftHeart project broadened to encompass both clinics. This was to coincide with a move by both clinics and a research lab to become integrated into a single cardiac prevention program. During early 1996, Clinic One, Clinic Two, and the Research Lab joined together within this umbrella organization (Heart Health
Chapter 4: Processes in the SoftHeart Case

Department; a pseudonym). The Heart Health Department (HHD) is funded by both the hospital and a heart center that supports heart disease clinics within the hospital. HHD is also funded by research grants and drug companies.

4.1.1. Clinic One

Clinic One is involved in the medical or clinical prevention of heart disease through traditional physician/patient visits and the prescription of diet, exercise, and medications to improve patient lipid values. Patients are referred to the clinic by general practitioners or cardiologists who believe their patients may have abnormal lipid values and a risk of heart disease. Clinic One’s physicians then analyze and prescribe treatment for the patient and provide information to both the referring physician and the patient. Dietitians analyze food intake information to assess dietary factors affecting lipids. Clinic One employs six physicians and two dietitians who assist referring physicians' treatment of patients at risk of heart disease because of abnormal lipid levels.

A useful way to visualize the past information system of Clinic One is to follow a typical patient's visit. After being referred to the clinic, the coordinator schedules the patient for a visit, and mails appropriate questionnaires. New patients provide basic family and personal information including a description of their medical history prior to their first visit. Both new and recall patients complete a food intake sheet prior to each visit. Upon arrival at the clinic, all patients are weighed, have a blood sample drawn, and are seen by one of the

---

88 The term ‘lipid’ is a more general term for ‘cholesterol’. Lipids comprise a range of greasy, waxy, and fatty compounds that are insoluble in water. Cholesterol is a ‘compound’ lipid used in the construction of cell membranes. An imbalance of good cholesterol (High Density Lipids - HDL) to bad cholesterol (Low Density Lipids - LDL) is associated with heart disease.
physicians. Approximately half of the patients see one of the dietitians as required. A physician spends about 20 minutes with a new patient, and about 8 to 10 minutes with a recall. Blood pressure, smoking, medical history, medication, and physical examination data are collected by the physician, and written in the clinical notes. A dietitian spends anywhere from 5 to 45 minutes with an individual patient. The Food Intake Sheet (FIS), filled out by the patient before the visit, is discussed and various calculations of current and ideal fat consumption, body weight, exercise, and calorie intake are manually calculated and written.

Upon making a diagnosis, the physician and the dietitian usually provide the patient with some verbal treatment information, supplemented with appropriate pamphlets and other documentation. The coordinator adds these notes and recommendation letters to the patients' charts, and updates a standalone database (EasySys, a pseudonym) with medication, weight, blood pressure, and smoking information. Two one-page reports are printed and stored in the patient chart for future visits; a lab/diet/weight/blood pressure and smoking spreadsheet, and a medication sheet.

At the end of the day, the physician prepares clinical notes for each patient, and the dietitians prepare dietary notes. The dietitian also makes brief informal notes in a “kardex” file. Once lab results are available (usually later than two days), the physicians will dictate letters for the referring physicians with recommendations for patient treatment.

New referring physicians’ information is entered into the computer. Phone calls from patients and referring physicians for lab information are answered by the coordinator using EasySys. EasySys stores patient demographics, lab, diet, and medication information about the patient gathered at each visit.
Clinic One is also involved in atherosclerosis research, focused largely on genetic causes. Subjects for the studies are recruited from the clinic, and basic research and clinical trials with medications are conducted. Most of the information and reporting for these research activities are held separately from the clinical care group in separate database programs and files.

4.1.2. Clinic Two

Clinic Two is involved in the lifestyle prevention of heart disease through a 6-week exercise program offered to patients at risk of, or recovering from, heart disease. In addition, clinic personnel provide weekly classes on stress, diet, exercise, smoking, and substance abuse. The clinic employs one physician who is also the director, two nurses, three exercise physiologists, a coordinator, and two clerks.

Patients are referred in a similar manner to Clinic One by general practitioners and cardiologists with patients recovering from heart disease. Patients are seen by the physician during an assessment clinic, who determines the patients' risk and need for treatment, and places them on a priority list for the program. Patients are eventually scheduled for a six-week exercise program involving two 1.5-hour classes of monitored exercise every week, and assigned a case manager based on their class time. Patients are checked again at mid-point through a diagnostic clinic, and at the end through an exit clinic. Stress test information gathered from an exercise test, determines the patient's capacity for oxygen at both the assessment and exit times. Currently, all information is handled manually in a patient file.

---

89 A disease that results in the fatty build-up of plaque in the artery walls causing eventual blockage of blood flow.
folder. Information is written several times in different formats to satisfy the needs of the various providers in the clinic.

Clinic Two is also involved in research activities focusing on various areas of heart disease rehabilitation and prevention. Information for this research is also collected separately, using patients recruited from their classes.

4.1.3. Description of SoftHeart and LipidLink

SoftHeart is an electronic patient record, designed to address three areas in order of importance: clinical (patient care), research, and administration. The SoftHeart software consists of various modules that involve three design and programming areas: database design, data input, and display (on screen and reports). Each of the modules are in various stages of development.

SoftHeart includes most modules required in a fully electronic patient record. The following lists the modules, and each module’s current completion status is indicated by a number, where “1” means database design only, “2” means database design and input, and “3” means database design input and display/reporting. Modules include lipids (3), lab results (3), medical history (2), family history (1), physical examination data (1-2), medications taken and prescribed (2), diet and exercise (3), smoking (3), alcohol (3), blood pressure (3), “stress test” (2), side effects (1), and quality of life (1) information. Administrative information is also gathered including patient demographics (3), referring physician information (3), and a

---

90 SoftHeart’s original focus was solely on clinical and patient care.
91 Screen pictures of SoftHeart are provided in Appendix E.
92 A “stress test” is an exercise test that measures the oxygen capacity of the patient.
scheduler (3) for class and/or appointment times. Finally, there are several modules that extend the information including a risk calculator that calculates 5 and 10 year risk of heart disease (3), a "modifiables" display panel displaying those items considered under the control of physician and patient (diet, exercise, lipids, medications, etc.) (3), and a graphing module to allow visual inspection of trends (2). The program is about 40% complete, but the modular design has allowed implementation of completed modules in the two clinics. Currently, medications, graphing, and connections with two hospital mainframe systems (administrative discharge tracking and lab medicine) are the main targets for the next months.

A view of all data to be eventually collected by SoftHeart is provided below:

**Figure 4-1: Data View of SoftHeart**
Chapter 4: Processes in the SoftHeart Case

The database structure attaches one-time data directly to the patient, and repetitive information gathered at visits to an event or encounter. Catalogues are also used extensively to allow data to be entered quickly and efficiently for diagnosis, exam, medical history, and side effects. Catalogues are also used to set parameters for abnormal values, to calculate food content in patient diet, and to calculate risk.

SoftHeart encompasses a number of design concepts. First, the software uses tabs to provide access to any information with the push of a button. Second, the software separates display from data entry by using panels of information. Depending on the user password, these panels allow data input at the push of an “edit” button, displaying a green editing region in the middle of the screen. A user’s password into the system determines the buttons that are displayed on the side and an asterisk indicates whether the panel can be edited by this user. The editing button is then enabled or disabled based on the button label assigned to this user. This feature provides for different levels of security into the data given the role of the individual. Third, each panel has its own program modules for displaying the screen and checking for data integrity, providing modular design. Fourth, SoftHeart’s design avoids delaying the user during display and data entry by placing messages off to the side and avoiding the need to press a key to clear, automatic saving, linking editing modules to allow entry to multiple panels without leaving the editing screens, and attachment of appropriate modules into edit screens where appropriate. For example, “Physicians...” and “Scheduling...” are attached in the patient input window to allow linking and scheduling of patients. SoftHeart currently uses over 30,000 lines of code.\textsuperscript{93}

\textsuperscript{93} Appendix G provides a programming guide to SoftHeart.
While the developer had been developing the software, the research arm of the two clinics, under the leadership of their director, developed the infrastructure named “LipidLink” (a pseudonym). The purpose of LipidLink is to provide a computer in every room, including the physician and dietitian consult rooms, to allow access and entry of information through SoftHeart. To accomplish this, the research group has managed to obtain funds from the Heart Center and a drug company to finance the hardware and network software.

SoftHeart has only recently begun to be used by the two clinics. Currently, most of the features of EasySys system are available and being used in SoftHeart. These include patient, lab, dietary, and physician data input, and reporting of lab sheets, patient labels, etc. There is also some evidence that new information is being collected, including exercise stress tests, smoking, and blood pressure. LipidLink now provides linkage to SoftHeart in almost every room, except for physician examination rooms. In addition, users are now starting to use the software to examine patient data and test graphing capabilities, and using direct Foxpro queries to query the data for research purposes from their terminals.

The following sections provide a stage-by-stage description of the ISD process of SoftHeart.

4.2. Project Emergence

Particularly important in understanding information systems development are the processes resulting in the initiation and creation of IS and IT. Project emergence outlines important processes in the initiation and creation of SoftHeart.

The SoftHeart project was started by a core group of researchers interested in health
care, information systems, medicine, and health economics. Particularly important in the project’s initiation were the goals and interests of the developer (Schon 1983; Turner 1987; Beath & Orlikowski 1994; Petroski 1994; Thomas 1994). His interest in integrating information systems and health care initiated the project. His background and motivation to research IT and its impact on health care organizations provided a innovation “subsidy” and an “innovation context” (Van De Ven 1986; Brooks 1987; Hirschheim & Klein 1989; King, Gurbaxani, Kraemer, McFarlan, Raman & Yap 1994; Thomas 1994). His database background from work experience equipped him with some skills for developing SoftHeart, access to the clinic’s information system (EasySys), and an interest in improving Clinic One’s databases and reporting mechanisms.

The developer’s inexperience with clinic settings required him to learn the clinical and heart disease setting in order to develop an IS intervention. This inexperience also gave the developer a perhaps naive and technical approach to design at the beginning of the project, and may have prevented an appropriate feasibility study of the site and hospital environment (Klein & Lyttinen 1985; Boland 1987).

---

94 Schon (1983) discusses the reflective practitioner and the effect of an individual’s background on interpretation and action. The practitioner’s understanding and background effects the course of development. Turner (1987) characterizes design as a goal, with no immediate procedures for attaining that goal. Beath & Orlikowski (1994) discuss the effect of methodology choice and approach on development. Petroski (1994) talks about the role of designers, their failures and background. Thomas (1994) shows that innovation in manufacturing comes from outside the management circle through technology developers.

95 The developer had little idea at the time what was the future of the project. He approached the problem from a technical perspective, somewhat unaware of the political context of medicine and Clinic One.

96 Van De Ven (1986) suggests that a central problem in innovation is the innovator’s ability to manage the project, and to fuse the interests of the different groups. Brooks (1987) discusses the complexity of software design through conformity and changeability of software, and the need for good designers. Hirschheim & Klein (1989) discuss four approaches of a developer in the course of development: functionalist, integrationist, radical structuralist, and neo-humanist. Each view depends on two dimensions: ontology (subjective/objective) and conflict (conflict/integration). Thomas (1994) shows that innovation in manufacturing comes from outside the management circle through developers.

97 Klein & Lyttinen (1985) describe the approach of the developer of SoftHeart during the early stages; a scientific background with a mixture of interpretive, and a naive and technological approach to development. Boland (1987)
As a result, the SoftHeart project was more an opportunity than a necessity \(^9\) (Basalla 1988; Petroski 1994). For the developer and his committee, these opportunities included an interest in health care and information systems, a need to find an interesting thesis topic, and a chance for three colleagues to carry out research together (the developer’s supervisor, one committee member, and the clinic physician). Thus, the project was driven by supplier and developers \(^10\) (Turner 1987; Attewell 1992; Thomas 1994).

Early proposals fused the interests of the different groups. The proposals focused on information “framing” \(^11\) and multimedia computers, and its effect on providing clear risk and treatment information to the patient that would produce changes in behavior and health outcome \(^12\) (Tversky & Kahneman 1981; Benbasat & Dexter 1985a & 1985b; Benbasat, Dexter & Todd 1986a & 1986b; Earl 1989; Cooper & Zmud 1990; Mohan, Holstein & Adams 1990; Tan & Benbasat 1990). These proposals described information technology’s
discusses the importance of context and background in interpreting information, including the purposes, motivations, and background.

\(^9\) At the same time, this naivete provided the developer with the “outsider” view of the events inside the clinic, giving him a more detailed and valid analysis of events and processes.

\(^9\) Basalla (1988) and Petroski (1994) both discuss the evolution of design. Technology arises from perceived failure and an “opportunity”, not necessity. It is the perceived failure of technology that drives innovation.

\(^10\) Turner (1987) discusses the reconciliation between human and technical in design. Technology embodies an idea from suppliers and developers. Attewell (1992) and Thomas (1994) say technology is a supply side phenomena, where knowledge is infused into the organization by suppliers both outside and inside the organization.

\(^11\) “Framing” involves the systematic shift in an individual’s interpretation of similar statistical information that is worded differently. In one study, tank crews were told they either had a 50% chance of death or a 50% of living if they chose to drive their tanks down a path. Those given the “death” framing were less likely to drive the path than those given the “life” framing.

\(^12\) Tversky & Kahneman (1981), Benbasat & Dexter (1985a & 1985b), Benbasat, Dexter & Todd (1986a & 1986b), Earl (1989), Cooper & Zmud (1990) and Tan & Benbasat (1990) were all influential in the initial ideas of fitting graphical information presentation to the tasks of patient education. Mohan, Holstein & Adams (1990) document some of the problems with low public sector usage of EIS (Executive Information Systems); a set of systems similar to SoftHeart that use graphical and “drill down” capabilities to view information at general and specific levels.
'built-in' positive effects on patient outcomes, and that a clinic which develops and implements these technologies will show increased patient health (Kim & Michelman 1990).

The following table outlines the important processes in the project emergence stage.

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer Background</td>
<td>EP</td>
<td>Project conception</td>
<td>Jan 1990 to July 1992</td>
<td>Interest and experience</td>
<td>Developer</td>
</tr>
</tbody>
</table>

4.3. **Health Funders Skeptical**

Initial reactions to the IS and IT ideas were harsh and critical, as targeted stakeholder’s “talked back”. The skeptical reaction of health agencies toward the proposals caused a systematic reformulation and rethinking of the ideas. The funding agencies rejected the 'black box' of computing (Latour 1987; 1994). They doubted the ability of “framing” and multimedia computers to accurately inform patients about risk, change behavior, and improve patient outcomes. While some agreed that IT could lead to improved decision making and patient treatment, others suggested computers would destroy rapport between the

---

103 Kim & Michelman (1990) point out the naiveté of this approach in hospitals given the conflict between medical and administrative branches, and the lack of system integration resulting in problems with strategic IS use.

104 Latour's work provides some insightful analysis into the controversies of scientific “fact” and technology during the early years of development. The black box of both science and technology hide this controversy in the future when social divisions are healed and agreement is reached on the theory and technology.
provider and the patient (DeLone & McLean 1992; Franklin 1990; Postman 1992). For those who believed in the potential of IT to improve patient health, a number felt IT was too expensive and too weak to produce definite health outcomes.

In general, the proposals appeared to violate key assumptions about the health context (Dutton 1981; Markus 1981; Kling & Scacchi 1982; Markus 1983; Van De Ven 1986; Orlikowski 1992b). The funding agencies understood IT as only a "hardware box", and misunderstood the intervention's focus on the communication between patient-and provider. Future proposals would focus less on the technology, and more on the communication of information to overcome these concerns (Van De Ven & Polley 1992). In addition, a number of health researchers commented that we would be seen as outsiders by the health agencies, and that overwhelming credentials and proposal strength would be required to obtain funding from these groups (Markus 1981; Markus 1983).

The following table summaries the important processes in this stage:

---

105 All three authors point to the problem of technology when applied inappropriately to a task, and Franklin (1990) and Postman (1992) discuss the divisive nature of current technology.

106 Certain "technologies" are perceived more strongly as imperatives in the health field, including medications and surgery. Information ‘treatment’ appeared to be too ambiguous to these groups.

107 The issue around whether computers were inherently harmful to the task and environment of health care raises questions about the context’s goals, assumptions and methods. Dutton (1981) demonstrates the important of context in computer’s use and rejection. Markus (1981) provides insights into this case’s structural power of the health funding network and the political setting around technology use in health care. Kling & Scacchi (1982) demonstrate the importance of web models and “web analysis”. Markus’ (1983) comments that technology must meet the structural powers in an environment to succeed, and a need for thorough analysis of political setting. Van De Ven’s (1986) managing attention and converting ideas into currency is an important insight into this process. Orlikowski’s (1992b) comment about the premises of computer technology and an incongruence between setting and technology, may explain the conflict with the health care setting.

108 Learning of the patient-provider and health care settings was inhibited by the search for funding, similar to the case in Van De Ven & Polley (1992).

109 Similar to Markus (1981), the development team encountered the structural power and assumptions about funding and IT research in medicine. Similar to Markus (1983), a thorough analysis of the political setting for IT usage was important, and system ideas were rejected by these funding agencies.
Table 4-3: Processes in Health Funders Skeptical

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Funders Skeptical</td>
<td>EP/TI</td>
<td>Project Conception, infrastructure</td>
<td>Jan. 1993</td>
<td>Surprise and struggle</td>
<td>Developer, health agencies, committee</td>
</tr>
<tr>
<td>Questioning computer and framing</td>
<td>EP/ST</td>
<td>Project Conception, data elements</td>
<td>Feb. 1993</td>
<td>Doubt</td>
<td>Developer, committee</td>
</tr>
</tbody>
</table>

4.4. **Our Response**

The core development group responded to the initial reaction of targeted stakeholder through a series of learning and action responses. The concepts and the information system details were adjusted and rewritten, based on an early exploration of the epidemiological and medical literature. The IS interventions were increased to develop a stronger experimental condition to satisfy the health agencies’ need for guaranteed results \(^\text{110}\) (Markus 1981; Kling & Scacchi 1982; Latour 1987; Orlikowski 1992b; Van De Ven & Polley 1992). The proposals also removed costly computing technology and focused on printed reports \(^\text{111}\) (Franz & Robey 1984; Van De Ven 1986). The purpose was to gain funding for the project’s core idea: improved information collection and presentation will improve patient outcomes.

\(^{110}\) This was an attempt to deal with experimental assumptions of funding agencies (Markus 1981). The implicit “discrete entity” view of technology did not work with the agencies (Kling & Scacchi 1982). The group was trying to close the black box using rhetoric produced rejection (Latour 1987). The proposals discussed the technology’s use toward communication between patient and provider, and a change in focus away from the technology per se (Orlikowski 1992b). The continuing attempts to satisfy the agencies led to a lack of detailed learning (Van De Ven & Polley 1992).

\(^{111}\) Rational carries the foreground while political in the background of the proposals (Franz & Robey 1984). This was an attempt to manage people’s attention, to create an innovation context, and to produce an authentic system to match expectations was required (Van De Ven 1986).
The developer also talked with clinic staff to determine feasibility and acceptability of plans (Ives & Olson 1984; Baronas & Louis 1988). The purpose was to understand clinic one’s current systems (Earl 1989), and to promote developer learning (Newman & Noble 1990). A structured analysis and system design document was produced (Coad & Yourdon 1991; Kendall & Kendall 1992).

Finally, the developer’s expanded reading of the heart disease, patient education and health promotion literature increased his awareness of the complex path between IT and patient outcomes, and opportunities for IT (Nadler 1982; Schon 1983; Franz & Robey 1984; Turner 1987; Earl 1989). In addition, his reading of this literature allowed him to communicate with both health agencies and clinic personnel (Boland 1985, 1990).

The following table summarizes these processes:

---

112 There was an assumption that user involvement was important, and the software was to be driven by user needs (Ives & Olson 1984). One goal of involvement was to restore a sense of control to the clinic staff, and to provide understanding of the work environment (Baronas & Louis 1988).

113 The purpose was to identify strategic aspects, and to determine what IT and IS already existed in the clinic (Earl 1989).

114 Both learning and conflict with the physicians were generated during the interviews, created by a lack of time and misunderstanding of the developer’s reasons for questioning (Newman & Noble 1990).

115 Object oriented analysis (Coad & Yourdon 1991), and Data Flow Diagrams (DFDs) and Entity Relationship Diagrams (ER) (Kendall & Kendall 1992) were produced.

116 The complexity involved in changing patient behavior emerged in the literature, as the developer red the health promotion literature (Nadler 1982). The developer used “scenarios” and reflective thought to test the feasibility of various IS interventions (Schon 1983). Rational approaches yielded complexity in patient/provider interaction, and political realities were being discovered in the background (Franz & Robey 1984). Goals were set to address patient education and communication, but process of reaching these goals through IT and IS were fuzzy and uncertain (Turner 1987). Nevertheless, the idea was to identify the strategic opportunities for IT in the clinical setting (Earl 1989).

117 Understanding of the language and the use of critical theory and hermeneutics to understand patient and physician information allowed detailed understanding of the organization (Boland 1985). Also, the developer’s background as a chronic patient allowed some ideas and expertise to be added (Boland 1990).
Table 4-4: Processes in Our Response to Health Critiques

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consult staff more closely</td>
<td>OI</td>
<td>System objectives and functionality</td>
<td>Feb. to Oct. 1993</td>
<td>Interview and discussions</td>
<td>Developer, clinic personnel, MBA students</td>
</tr>
</tbody>
</table>

4.5. **Old and New Environmental Shocks**

Once again, a period of reaction from targeted stakeholders appeared during this stage of development. In addition, environmental shocks occurred that changed the course of development.

Just as the project gained momentum at the end of 1993, several environmental shocks created confusion and uncertainty about the project’s continued viability. When Hospital One closed and the Clinic One was forced to move in early 1994, there was a fear that the clinic might close forever and the project would die (Dutton 1981; Lyytinen & Hirschheim 1987). Doubt about the project’s continuation remained when the clinic was moved to Hospital Two,

---

118 Change in hospital settings and support networks threatened to make SoftHeart irrelevant or unsustainable. This is similar to Dutton’s (1981) work examining the demise of a land planning system after governments changed. The project was pushed into the background, and failure loomed (Lyytinen & Hirschheim 1987).
Chapter 4: Processes in the SoftHeart Case

and the SoftHeart project was pushed into the background as Clinic One struggled with the move and the new setting (Van De Ven 1986).

In addition, funders remained skeptical, especially with the now complex interventions designed to address the complicated link between information and patient outcomes. Concern focused on the "contamination" of control groups by experimental procedures, and therefore the inability to measure outcomes (Markus 1981; Nadler 1982; Markus 1983; Latour 1987; Lyytinen & Hirschheim 1987).

Finally, by addressing the health issues and the removal of expensive information technology, the university MIS department reacted to the lack of advanced computers in the project (Markus 1983; Orlikowski & Robey 1991). The constraint influenced the course of the project because of the developer's need for IS research material (Dutton 1981).

The following table summarizes the processes in the environmental shock stage:

---

119 The project lost focus and interest amongst clinic personnel, similar to Van De Ven's (1986) "managing attention" and conversion of an idea into "currency".

120 The interdisciplinary nature of the project caused health, IS and interdisciplinary groups to reject our proposals (Markus 1981, 1983). The complexity of dealing with social and psychological realities of patient care continued to effect the project direction (Nadler 1982). The black box of IT continued to remain open to controversy over usage and content within the health care field (Latour 1987). Failure continued to loom over the project with the failure to acquire funding and resources (Lyytinen & Hirschheim 1987).

121 The structural power and boundary spanning activities of the developer offended the IS group at the university (Markus 1983; Rogers 1995). Norms, resources, institutional norms of research were affecting the course of the project (Orlikowski & Robey 1991).

122 The developer was caught between two settings; health wanted guaranteed health outcomes and MIS wanted advanced technology (Dutton 1981).
Table 4-5: Processes in the Environmental Shocks Stage

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hospital One closes</td>
<td>EP</td>
<td>Project conception</td>
<td>Jan - Mar 1994</td>
<td>Surprise, doubt and confusion</td>
<td>Developer, Clinic One, government</td>
</tr>
<tr>
<td>Funders continuing skeptical</td>
<td>EP/ST</td>
<td>Infrastructure</td>
<td>Jan - Mar 1994</td>
<td>Surprise and struggle</td>
<td>Developer, committee, health agencies</td>
</tr>
<tr>
<td>MIS' need for advanced computer study</td>
<td>EP/ST</td>
<td>Infrastructure</td>
<td>March 1994</td>
<td>Struggle</td>
<td>Developer, organization, MIS faculty</td>
</tr>
</tbody>
</table>

4.6. Decisions to Move Ahead

Processes during this stage highlight the action and learning of the developer to cope with opportunities and problems identified in the previous stage of development. As the dust cleared in April of 1994, the developer tested and redeveloped the core concept of SoftHeart to deal with the constraints imposed by health funders and the university MIS department. He used his own time and labor to develop “add-on” programs to EasySys, Clinic One’s standalone patient record, including the printing of graphical and risk information for patient/provider consults. In this respect, he was trying to leverage and capitalize on a data resources already available at the clinic (Earl 1989). In addition, the developer hoped these “add-ons” would provide evidence to health and other funding agencies that the project was

---

123 This was an attempt to leverage existing IT and IS resources inside the clinic (Earl 1989).

Foxpro was selected as the software platform. This selection was based on experience, intuition, limited assessment of the EasySys program, and experimentation \(^{125}\)(Brooks 1987; Turner 1987; Thomas 1994). Costing only $135 dollars, the package could be discarded if it failed to meet current and future needs \(^{126}\)(Schon 1983; Landauer 1991).\(^{127}\)

Learning the Foxpro software and building a general approach to program design required time and effort. However, this learning allowed the designer to explore the potential and new possibilities for extending information presentation as the developer explored the software’s capabilities \(^{128}\)(Carrol & Mack 1984; Van De Ven & Polley 1992; Hammer & Champy 1993).\(^{129}\)

\(^{124}\) This was a political maneuver to satisfy MIS, to avoid funding agencies’ concern about expensive programming and computer technology; and to keep the project moving (Markus 1981; Van de Ven 1986). Adding and modifying the existing EasySys software points to the use of legacy systems to evolve and create newer systems by using the older technology to generate ideas (Basalla 1988). The change in direction also moved the project away from marketing and into idea generation and learning (Van De Ven & Polley 1992). The maneuver also demonstrated the important push of the supplier and developer in producing and selling innovation (Thomas 1994).

\(^{125}\) The project was growing software out of the current system (Brooks 1987). An attempt was made to reconcile human and technical using intuition and experience (Turner 1987). Selecting the tools for innovation, and working in the background suggests a supplier led development and a political process to gain legitimacy in the clinic (Thomas 1994).

\(^{126}\) Thought experiments were conducted working with current and future scenarios (DBF files etc.) (Schon 1983). Mock ups were developed to learn, and thrown away if inappropriate (Landauer 1995).

\(^{127}\) It was doubtful whether further planning and search period would have solved this problem because the needs of the SoftHeart system were discovered throughout the project.

\(^{128}\) Users were involved in the learning process by direct design and discussion about design (Carrol & Mack 1984). Now that funding worries were left behind, the learning cycle began (Van De Ven & Polley 1992). Identification of processes and information technology that could enhance or transform those processes was the goal (Hammer & Champy 1993).

\(^{129}\) This process continues to this day.
The developer also decided, based on a belief that user involvement would be essential to system success (Baroudi, Olson, & Ives 1986), that he needed increased access and involvement from the clinic personnel to develop and test IS ideas. It was also important to garner the interest and involvement of physicians, the director, and the coordinator to get an understanding of EasySys, and to gather suggestions and commitment to the intrusive use of computing technology. He also needed to learn about their organization and tasks (Van De Ven 1986; Earl 1989; Newman & Noble 1990; Attewell 1992). As a result, he began to observe patient-provider discussions and clerical tasks. A prototyping approach was also used to test ideas and search for an integrated understanding and set of IS needs among clinic personnel (Boehm 1988; Gould 1988; Newman & Noble 1990; Cooper & Zmud 1990; Henderson & Kyng 1991; Wynn 1991; Landauer 1995).

To have a powerful effect on patient outcomes, the software add-ons needed to provide "value-added" tools for information presentation (Earl 1989). Therefore, early prototypes focused on risk calculation and graphing of information to effect patient provider communication. The developer also continued his learning about the medical and health

---

130 Ives, Baroudi & Olson (1986) discuss the general and often incorrect assumption that user involvement will guarantee system success.

131 Managing attention, getting people's interest, and a need to learn about the organization drove the desire for user involvement (Van De Ven 1986). Identification of strategic areas for IT usage and assessment of resources was another hidden assumption of user involvement (Earl 1989). User involvement was perceived as an interaction between users and specialists for learning and conflict resolution (Newman & Noble 1990). Learning and the barriers to learning about IT were implicitly being addressed in discussions (Attewell 1992).

132 Spiral development was believed to be the best way to deal with organizational complexity and to identify promising future IT projects (Boehm 1988). The result was a focus on users, iterative design, and empirical testing of ideas (Gould 1988). User involvement was an interaction between users and specialists for learning and conflict resolution (Newman & Noble 1990). The purpose was the identification of appropriate tasks that could use information technology (Cooper & Zmud 1990). Tailoring and an understanding of the work environment was believed to be essential to develop a good information system (Henderson & Kyng 1991). The developer took the nuances of practice seriously through direct observation of health providers (Wynn 1991). The developer employed user interface design and testing with users, and formative design evaluation (Landauer 1995).
promotion literature to learn what technologies and programs had worked or failed with patients in previous studies (Weitz 1990; Orlikowski & Robey 1991; DeLone & McLean 1992). The following summarizes the processes during this stage:

Table 4-6: Processes in the Move Ahead Stage

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add-ons to EasySys software</td>
<td>EP/OI/ST</td>
<td>Infrastructure, program, processing, reports, interface</td>
<td>April 1994</td>
<td>Learning and understanding</td>
<td>Developer, Clinic One personnel</td>
</tr>
<tr>
<td>Selecting software platform</td>
<td>EP</td>
<td>Infrastructure, program</td>
<td>May 1994</td>
<td>Search and comparison</td>
<td>Developer, software suppliers</td>
</tr>
<tr>
<td>Learning the Foxpro software</td>
<td>EP/OI</td>
<td>Infrastructure, program, processing, interface, display, data structure</td>
<td>June 1994 to June 1995</td>
<td>Learning, frustration, reading, learning</td>
<td>Developer, suppliers, textbooks</td>
</tr>
<tr>
<td>Prototyping approach adopted</td>
<td>OI/EP</td>
<td>Interface and data</td>
<td>July 1994 to present</td>
<td>Creation and reaction</td>
<td>Developer, clinic personnel</td>
</tr>
<tr>
<td>Focus of risk calculation and graphing</td>
<td>OI</td>
<td>Functionality and interface</td>
<td>August 1994</td>
<td>Value added functions and project positioning</td>
<td>Developer, software</td>
</tr>
<tr>
<td>Learning about medical and health promotion</td>
<td>ST/OI</td>
<td>System objectives and functionality</td>
<td>August 1994 to present</td>
<td>Learning, understanding, addressing issues</td>
<td>Developer, environment (literature)</td>
</tr>
</tbody>
</table>

133 Focus on strategic and "value added" tools for patient education (Earl 1989).
134 In anticipation of researching this question, the developer began to sit-in on a number of provider-patient visits to assess current communication patterns.
135 Societal values and policy choices determined within the context of health care and health promotion were to shape technology (Weitz 1990). Learning would address the institutionalization of technology after implementation and the need for the content of the system to address the constraints and opportunities identified in the health promotion literature (Orlikowski & Robey 1991). Task technology fit and the attempt to use IT to achieve an organizational impact were employed (DeLone & McLean 1991).
136 See appendix C for a literature review of the heart disease and theories of health behavior.
4.7. **Presentation Prompts Interest and Constraint**

Processes during this stage highlight the complex interaction between developer and users during SoftHeart’s unveiling. The presentation of the software to the clinic personnel on September 29th produced both intended and unintended consequences. Clinic personnel were excited about the project (Newman & Noble 1990). In addition, the project exposure created an understanding of the SoftHeart project amongst clinic personnel, and sparked involvement (Ives & Olson 1984; Baroudi, Olson, & Ives 1986; Baronas & Louis 1988; Robey, Farrow, & Franz 1989; Hirschheim & Klein 1989).

To the developer’s surprise, this involvement produced a barrage of numerous and conflicting suggestions. For example, the risk calculator used “total cholesterol” to calculate risk, which raised concerns amongst some physicians. Other physicians stated that the calculator was relatively well-known in the general practitioner community, and was ideal for communicating information to that group. Development and usage of the risk calculator was more complex and controversial than expected (Nadler 1982, Schon 1983; Van De Ven

---

137 Excitement, learning and user involvement were generated (Newman & Noble 1990).

138 Especially interesting was the general agreement that the graphing module would provide universal benefit to patient education, not provider education (Benbasat, Dexter & Todd 1986a; Benbasat, Dexter & Todd 1986b; Tan & Benbasat 1990).

139 Interest and understanding were sparked by the presentation (Ives & Olson 1984). Information satisfaction and usage were the goals of the presentation (Baroudi, Olson, & Ives 1986). The developer was attempting to build a sense of control amongst the clinic personnel to achieve agreement (Baronas & Louis 1988). Participation led to influence, conflict, and some conflict resolution (Robey, Farrow & Franz 1989). What began as an attempt to integrate users generated structural divisions that would lead to future conflict (Hirschheim & Klein 1989). Structural conflict was suggested by the disagreement over risk calculator. Whose side does the developer take? Complexity of design, and the possibility of a radical alteration of the organization could result from the introduction of the risk calculator (Harrington 1991). Ideas and development now being taken over by users in the organization, after two years in the developer’s hands (Thomas 1994).

140 Complexity of design illustrated by the disagreement over the risk calculator (Nadler 1982). Context “talks back” about the risk calculator, including the feasibility and controversy (Schon 1983). Conflict resolution and conflict creation occurred considering the use of the calculator (Robey, Farrow, & Franz 1989).
Chapter 4: Processes in the SoftHeart Case

1986; Boland 1987; Turner 1987; Robey, Farrow, & Franz 1989; Hirschheim & Klein 1989; Harrington 1991; Thomas 1994). Many of the suggestions also indicated that the comfort of the users was as equally important in the design as patient outcome (Bostrom & Heinan 1977). The developer’s ‘scientism’ view of ISD was beginning to change (Klein & Lyytinen 1985; Boland 1987).

The diversity of these wants and desires also created a dramatic expansion and stretching of the SoftHeart concept (Van De Ven 1986). Each group saw the technology as providing something for its specific needs, falling roughly into clinical, administrative, and research purposes (Kling 1980; Kling & Scacchi 1982; Markus 1983; Franz & Robey 1984; Newman & Noble 1990). Physicians focused on the potential of the database to store new data, improve information access, and improve information presentation for clinical and research purposes. Dietitians believed the system would provide computer resources in the consult room for immediately calculating patient dietary intake, and printing communication

141 Sociotechnical design and the consideration of worker quality of life were being considered as important as technological development (Bostrom & Heinan 1977).

142 “Scientism” focuses only on technology development and deployment, and is criticized in the literature (Klein & Lyytinen 1985; Boland 1987).

143 The management of the core concept is important in innovation (Van De Ven 1986).

144 Segmented institutionalism suggests there is a nondeterministic relationship between technology and context during development. Computers are seen as a package surrounded by a social and political setting. In the SoftHeart case, each group saw the technology as something specific and different (Kling 1980). Web models and social leverage were prevalent. Unfortunately, only the narrow boundaries of technology were originally considered by the developer. With so many different and new groups, the complexity of handling these views became complex (Kling & Scacchi 1982). Political perspectives on information systems illustrate that each group saw the technology as something specific and different, and the satisfaction of each group was important for future system survival (Markus 1983). Rational and political perspectives on the risk calculator suggestions, originally sounding like a rational debate, but in many cases, becoming political (Franz & Robey 1984). Conflict resolution, learning and political action appeared, as each group saw the technology as something specific and different, and measured its success by the power and prestige it could offer them individually (Newman & Noble 1990).

145 Every physician was impressed with the graphing capabilities of SoftHeart for patient education and research analysis. It was a major selling point of the software.
messages for patients. The coordinator would see the project as an opportunity to produce a windows-based version of EasySys, with a few additional improvements for improve administrative efficiency.\footnote{146}

One particular group interested in SoftHeart was the research group and its director. The merger of SoftHeart with the research group’s vision of an integrated wide-area network (LipidLink) produced both opportunities and constraints \footnote{147}(Kling 1980; Markus 1983; Orlikowski & Robey 1991). The opportunities included development of the computing infrastructure, and additional funding and human resources for the project. The first meetings between the research group and the developer, however, also revealed new requirements and constraints on SoftHeart’s design, including the research group’s desire to use OS/2 as the operating system, to build strict data entry checking, and to develop extensive documentation.

The following tables summarizes the processes appearing in this stage:

\footnote{146 When clinic two entered the project in March of 1995 they would see the SoftHeart project as an opportunity to move away from laborious manual tasks of recording, reporting, and counting patient information. They also saw the potential of the program to provide evaluations of their exercise program to themselves, hospital, and a government group skeptical of their program’s efficacy.}

\footnote{147 Segmented institutionalism describes the interactions and organizational politics. The SoftHeart system encompassed many old and new groups, producing opportunities and constraints (Kling 1980). The power structure and political nature of IS were reflected in the new constraints (Markus 1983). Norms, power, and resource within the organization produced constraints (Orlikowski & Robey 1991).}
Table 4-7: Processes in the Interest and Constraint Stage

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presentation of Prototype</td>
<td>EP</td>
<td>Software functionality</td>
<td>Sep. 29 1994</td>
<td>Generate interest and direction</td>
<td>Users, Developer, Task, organization</td>
</tr>
<tr>
<td>Barrage of user suggestions</td>
<td>EP</td>
<td>Functionality, interface, reporting</td>
<td>Sep. 1994 to present</td>
<td>User Involvement and struggle to integrate</td>
<td>Developer, clinic personnel</td>
</tr>
<tr>
<td>Discovery and joining of research group to LipidLink</td>
<td>EP/ST</td>
<td>System functionality, infrastructure, input, processing, data structure, program</td>
<td>Oct. 1994</td>
<td>Confusion, Learning, and painful learning</td>
<td>Developer, research group (users), environment</td>
</tr>
</tbody>
</table>

4.8. Conflict Erupts

Processes during this stage highlight the structural power and conflict that erupted during information system development. In addition to the new constraints and requirements from the research group, the developer’s new relationship with the research group reopened a schism between this group and the coordinator that had been unknown to him.

A four-year history of the conflict between the research group and the coordinator had started when the research group’s software (CholeSys; a pseudonym) failed to replace EasySys 148 (Markus 1981; Kling & Scacchi 1982; Hirschheim & Klein 1989; Orlikowski 1991; Thomas 1994). CholeSys’ strict data entry annoyed and challenged the coordinator’s

---

148 Political power struggle over CholesSys surfaced. The hiring of consultant by coordinator killed the project (Markus 1981). Web models of computing and the wider context and history of organizational conflict effected the project (Kling & Scacchi 1982). Radical structuralist and neo-humanist approaches to development appeared (Hirschheim & Klein 1989). Control mechanisms around computing and its control on other groups was central in SoftHeart’s development (Orlikowski 1991). Supplier led innovation from inside the organization was rejected by powerful forces maintaining the status quo (Thomas 1994).
freedom to control her data entry work \textsuperscript{149}(Mohr 1987). Written in dBase and recently transported into Foxpro, CholeSys contained a number of software bugs. The hiring of an "independent" consultant by the coordinator and Clinic One produced a report that criticized the maintenance and structure of the program. CholeSys was never implemented, and the research group believed the coordinator had killed the project \textsuperscript{150}(Markus 1981; Markus 1983).

The conflict between the research group and the coordinator would not have affected the development of SoftHeart if the project would have left EasySys alone. In many respects, this was the preferred strategy because programming would have been kept to a minimum. However, the developer discovered a number of problems with EasySys that affected the project's focus on risk calculation and graphing. These problems included missing data for risk calculation, missing fields for important risk data, EasySys' locking of files preventing information sharing, and the inaccessibility to EasySys' source code \textsuperscript{151}(Lyytinen & Hirschheim 1987; Cooper & Zmud 1990; Orlikowski 1991; Orlikowski 1992). As a result, the developer began building EasySys' data-entry functionality into SoftHeart in order to achieve the strategic benefits of risk calculation and graphing \textsuperscript{152}(Earl 1989).\textsuperscript{153}

\textsuperscript{149} Two technologies wrapped in a conflict over control of information entry and delivery (EasySys and CholesSys). The same technology (SoftHeart) produced totally different attitudes amongst different parties (Mohr 1987).

\textsuperscript{150} System rejected because it failed to satisfy structural power interests in the organization (Markus 1981; Markus 1983).

\textsuperscript{151} EasySys perceived by the developer as incapable of satisfying the present goals of health promotion and research (Lyytinen & Hirschheim 1987). There was a task-technology fit problem (Cooper & Zmud 1990). There was a desire of the research group to control the work of the coordinator using CholesSys (Orlikowski 1991). EasySys, with its inability to be modified, no longer supported in a changing social and technological environment (Orlikowski 1992).

\textsuperscript{152} There were obstacles to the strategic sharing of information with EasySys (Earl 1989).

\textsuperscript{153} A single structural change to the database for new fields meant the old lipid program would be unable to run, because it did not support the data entry or reporting of that information. In addition, because the program locked the files during use, a new program to perform the new data entry and reporting of the data field could not use the data.
The developer’s recent involvement with the research group, the coordinator’s general dislike of outsider intrusion, and the developer’s efforts to change Clinic One’s information system caused the coordinator to demand that SoftHeart’s interface be identical to EasySys. Later, these and other demands would become unfriendly and produced ever-changing obstacles to further development as the coordinator refused to work with the developer (Hirschheim & Newman 1991). User involvement and attempts to integrate the project goals with hers and those of the research group led to obstruction and derision (Turner 1987; Robey, Farrow, & Franz 1989; Newman & Noble 1990; Thomas 1994).

The following table summarizes the processes appearing in this stage:

---

154 User involvement and integration beneficial. Resistance perceived as dysfunctional. Politics needs to be a concern of the developer (Hirschheim & Newman 1991).

155 Goals were ill defined, and the means to those goals were fuzzy and obscured by the developer’s inability to distinguish legitimate demands from user resistance. Design complexity and new issues emerged (Turner 1987). Unresolved issues of control and direction in the clinic amongst the clinic personnel (Robey, Farrow & Franz 1989). Conflict and learning about requirements, opportunities and structural conflict occurred (Newman & Noble 1990). Political struggle emerged between the administrative staff and the developers over the supplier-led and developed system (Thomas 1994).
Chapter 4: Processes in the SoftHeart Case

Table 4-8: Processes in the Conflict Stage

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery of previous software failure</td>
<td>EP/ST</td>
<td>Project conception</td>
<td>Oct. 1994</td>
<td>Surprise and bitterness</td>
<td>Research group, developer, coordinator</td>
</tr>
<tr>
<td>Problems with EasySys</td>
<td>ST/OI</td>
<td>Data management, data elements</td>
<td>Oct. to Dec. 1994.</td>
<td>Frustration, need to move to new software</td>
<td>Developer, coordinator, old software</td>
</tr>
<tr>
<td>Starting to incorporate legacy systems</td>
<td>OI/TI</td>
<td>Interface, display, input, reporting</td>
<td>Jan 1995 to present</td>
<td>Conflict</td>
<td>Developer, coordinator, organization</td>
</tr>
<tr>
<td>Conflict</td>
<td>EP</td>
<td>Functionality, infrastructure, interface, input</td>
<td>Jan 1995 to March 1995</td>
<td>Time consumption and exhaustion</td>
<td>Developer, coordinator, users, organization</td>
</tr>
</tbody>
</table>

4.9. Decision to Act

Processes in this stage highlight the response of the developer team to conflict and politics appearing during the previous stage. Conflict with the coordinator, ever changing and increasing “requirements” from discussions with various individuals, and an offer to move to another project prompted the developer to write a series of conditions for his continuation with the project. This caused a sudden and unexpected chain of events over the next week leading to the removal of the coordinator.

In late March, the developer concluded that internal problems within the clinic were the cause of the project’s slow progress. From the developer’s perspective, the project was
failing to achieve development goals (Lyytinen 1987; Lyytinen & Hirschheim 1987; Hirschheim & Newman 1991; Myers 1994a). Having received an offer to study the EMPOWER project earlier in the week, the developer now had the leverage for action (Markus 1983).

The threat to withdraw prompted a series of structural changes as a number of clinic personnel were motivated to keep the project alive (Hirschheim & Klein 1989). The developer’s list of conditions included equal power with the coordinator and director over administrative matters in the clinic, clinic acceptance of implementation deadlines, and a restriction of programming to areas of direct importance to patients (Van De Ven 1986).

Meetings with the lead physician and the director reconfirmed their commitment to the project. The developer told the director that he needed the source code for EasySys to modify the program to accommodate his research goals. Later that week, the director requested the EasySys’s source code from the coordinator. The coordinator’s refused to contact the programmer, leading to a confrontation between her and the director (Markus 1983). At a prototype demonstration at the end of March, the physicians’ realized that the developer knew the EasySys software and had the capability to replace it. Consequently, the

---

156 The change required in technology, organizational structure, and language context was not occurring (Lyytinen 1987). According to the developer’s view, development was failing (Lyytinen & Hirschheim 1987). The belief in the universal success of user involvement was rejected (Hirschheim & Newman 1991). Certain people and groups stood to lose or gain from SoftHeart, and the developer was caught in the middle (Myers 1994a).

157 Power, politics, a lack of commitment by important groups, and an offer to work elsewhere provided the leverage for action (Markus 1983).

158 Structural conflict and a threat to withdraw caused political turmoil (Hirschheim & Klein 1989).

159 The developer was trying to convert ideas into good currency, and he required control over deadlines and important administrative matters (Van De Ven 1986).

160 A power struggle demonstrated the structural power imbalance between the director and the administrative arm of the organization (Markus 1983).
physicians met after the demonstration, and the coordinator was fired\(^{161}\) (Markus 1983; Harrington 1991).\(^{162}\)

The following summarizes the processes involved in this stage:

**Table 4-9: Processes in the Decision to Act Stage**

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization­­al aspects causing project failure</td>
<td>EP</td>
<td>Project direction</td>
<td>Jan to March 1995</td>
<td>Interest and worry</td>
<td>Developer, environment, clinic personnel</td>
</tr>
<tr>
<td>List of demands given</td>
<td>EP</td>
<td>Project continuation</td>
<td>March 22</td>
<td>Ultimatum</td>
<td>Developer, director, coordinator</td>
</tr>
<tr>
<td>Director confronts coordinator for the source code</td>
<td>EP</td>
<td>Project setting</td>
<td>March 23</td>
<td>Conflict</td>
<td>Develop and coordinator</td>
</tr>
<tr>
<td>Presentation leads to firing</td>
<td>EP</td>
<td>System goal/objective, system functionality</td>
<td>March 27</td>
<td>Force</td>
<td>Developer, coordinator, director, users</td>
</tr>
</tbody>
</table>

4.10. *Project Builds Momentum*

Processes identified during this stage of development highlight the momentum the project gained after the conflict phase was resolved. The project moved into a new stage of development in early April of 1995. After the firing of the coordinator on March 31 1995,

\(^{161}\) The firing illustrated a political struggle that had been deeply embedded in the organization (Markus 1983). Intervening in the organization with the SoftHeart project had upset a four year draw between the various groups, and left the organization without the resources of the coordinator (Harrington 1991).

\(^{162}\) Many clinic personnel recounted stories about long-standing conflict and political fighting of the coordinator with clinic personnel, including three individuals that had left because of the conflict.
and with rising support and interest in the project from various internal and external groups, the project’s momentum increased.

The interest and involvement of the hospital’s MIS groups suggested new opportunities and constraints (Kling 1980; Dutton 1981; Markus 1981; Kling & Scacchi 1982; Franz & Robey 1984; Van De Ven 1986; Orlikowski & Robey 1991). These groups became important in either closing the project down or supporting its future.

The joining of Clinic Two provided an opportunity to test and develop SoftHeart in an organization focused more on the prevention of heart disease than Clinic One, with a currently manual information system and a more cohesive staff that Clinic One. As a result, the basic design of the software was reassessed, and resulted in structural changes in the data and software interface (Dutton 1981; Brooks 1987; Turner 1987; Argyris, Putnam, & Smith 1985).

During this period, a scheduling module for SoftHeart was considered. The decision to build a scheduler was the first attempt to deal with the unique needs of Clinic Two, and to address serious administrative problems in Clinic One affecting patient care (Van De Ven 1986; Earl 1989).

163 There was a wider web of resources and groups interested in joining the project (Kling 1980). The changed context at Hospital Two was very different from Hospital One (Dutton 1981). The developer was learning about the structural power and interest groups that would affect the project (Markus 1981). The web of computing and social leverage appeared (Kling & Scacchi 1982). Rational and political perspectives were appearing in the meetings with different groups (Franz & Robey 1984). There were problems keeping the project focus, and converting ideas into currency (Van De Ven 1986). Elements of structural power and adaptive structuration were illustrated (Orlikowski & Robey 1991).

164 The Second Clinic produced a dramatic change in the context of SoftHeart’s operation (Dutton 1981). SoftHeart was growing and adjusting to new pressures and opportunities (Brooks 1987). The increased complexity of means and goals resulted in a need to clarify boundaries (Turner 1987). Double loop learning was appearing (Argyris, Putnam & Smith 1985).

165 SoftHeart’s information system concept was being stretched (Van De Ven 1986). The Scheduler was an attempt to address an important area to patient care (Earl 1989).
In addition, LipidLink required major expenditures in hardware and software to achieve project goals. The search for funding by the clinic was a response to the developer's threat to leave and a possibility of the project failing without a serious commitment of resources (Van De Ven 1986; Lyytinen & Hirschheim 1987; Orlikowski & Robey 1991; Van De Ven & Polley 1992). As a result, the "web of computing" continued to expand (Kling & Scacchi 1982; Kling 1987) and the project gathered momentum and new and old individuals from the various dimensions (organization, user, environment, developers). The following summarizes the processes involved in this stage:

**Table 4-10: Processes in the Momentum Stage**

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hospital groups become interested</td>
<td>ST/EP</td>
<td>Functionality, infrastructure</td>
<td>Nov. 1994 to present</td>
<td>Cooperative and competitive ideas</td>
<td>Developers, hospital funding groups, MIS</td>
</tr>
<tr>
<td>Joining of Second Clinic</td>
<td>OI/EP</td>
<td>System functionality, infrastructure, data structure, program</td>
<td>March 30, 1995</td>
<td>High expectations, new ally</td>
<td>Developer, users in Clinic Two, environment</td>
</tr>
<tr>
<td>Decision to move into Administrative Areas</td>
<td>OI/ST</td>
<td>System Functionality</td>
<td>April 25, 1995</td>
<td>Strategic and assistance</td>
<td>Developer, clerks</td>
</tr>
<tr>
<td>Search for Funding and Resources</td>
<td>ST</td>
<td>Infrastructure</td>
<td>April 6, 1995</td>
<td>Optimism</td>
<td>Developer, research director, director of Clinic One</td>
</tr>
</tbody>
</table>

---

166 An innovation context was required to support development, and there was a need to manage whole-part relations (Van De Ven 1986). System success and failure depended on these resources. Failure now being redefined based on a full-scale electronic patient record (Lyytinen & Hirschheim 1987). There was an important need to address resources and organizational institutions to support the development of software and IT infrastructure (Orlikowski & Robey 1991). There was a renewed focus on funding and marketing, but learning was maintained (Van De Ven & Polley 1992).

167 Web models of resources and infrastructure to support new electronic patient record were required (Kling & Scacchi 1982). In addition, there was a need to define system boundaries (Kling 1987).
4.11. Development Quagmire

The increased momentum, however, combined with a series of new processes, created a bottomless list of requirements for SoftHeart. Soon after or perhaps because of the increased momentum, the developer became concerned with endless development and uncertainty. The increasing requirements and complexity of development reflected the increasing involvement of many diverse groups. These groups were diverse in their views of medicine, treatment, and technology in treatment. Reconciliation and identifying priorities was extremely difficult and not forthcoming from the groups themselves (Nadler 1982; Van De Ven 1986; Brooks 1987; Kling 1987; Turner 1987; Henderson & Kyng 1991).

The loss of key personnel involved in early design and support for the project reduced the project’s momentum, and the introduction of new members required relearning and discussion (Van de Ven 1986; Attewell 1992). Much time was consumed discussing and motivating new members.

An already fragile relationship between the developer and the research group was fractured by a series of computer crashes (Dutton 1981; Kling & Scacchi 1982; Lyytinen 1987). Keeping focused and managing attention became important but difficult (Van De Ven 1986). Social pressures from various groups stretched the project in numerous directions (Brooks 1987). Defining the boundaries of the system became difficult but essential (Kling 1987). Defining boundaries and setting priorities was obscured by a fuzzy identification of goals and means to achieving those goals using IT (Turner 1987). Increased pressure and dangers of tailoring the system to everyone’s needs increased (Henderson & Kyng 1991).

The need to reestablish awareness, rapport, trust and enthusiasm with new people consumed time and effort (Van De Ven 1986). Learning and key sources of information and project support were lost (Attewell 1992).

Computer crashes fractured a volatile relationship, changing the setting of development (Dutton 1981). The web of computing around technological operation failed (Kling & Scacchi 1982). The contexts around the technology and the understanding of that environment was uncertain and incomplete (Lyytinen 1987). The project was failing (Lyytinen & Hirschheim 1987).

---

168 Many groups and pressures from these groups increased complexity (Nadler 1982). Keeping focused and managing attention became important but difficult (Van De Ven 1986). Social pressures from various groups stretched the project in numerous directions (Brooks 1987). Defining the boundaries of the system became difficult but essential (Kling 1987). Defining boundaries and setting priorities was obscured by a fuzzy identification of goals and means to achieving those goals using IT (Turner 1987). Increased pressure and dangers of tailoring the system to everyone’s needs increased (Henderson & Kyng 1991).

169 The need to reestablish awareness, rapport, trust and enthusiasm with new people consumed time and effort (Van De Ven 1986). Learning and key sources of information and project support were lost (Attewell 1992).

170 Computer crashes fractured a volatile relationship, changing the setting of development (Dutton 1981). The web of computing around technological operation failed (Kling & Scacchi 1982). The contexts around the technology and the understanding of that environment was uncertain and incomplete (Lyytinen 1987). The project was failing (Lyytinen & Hirschheim 1987).
Chapter 4: Processes in the SoftHeart Case

1987; Lyytinen & Hirschheim 1987). The finger-pointing and harsh reactions by both the developer and the research group threatened to create a schism between the two groups.

As a result of an offer to move once again to another project, the developer threatened to quit 171(Markus 1983; Attewell 1992; Keil 1994).

The following summarizes the processes in involved in this stage:

Table 4-11: Processes in the Development Quagmire Stage

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explosion In Requirements</td>
<td>EP</td>
<td>All levels</td>
<td>May to August 1995</td>
<td>Optimism, confusion, complexity</td>
<td>Users, Clinic One and Two, developer</td>
</tr>
<tr>
<td>Loss of Key Personnel</td>
<td>EP</td>
<td>Functionality, infrastructure, program, interface, data</td>
<td>March to Sep. 1995</td>
<td>Surprise and sadness</td>
<td>Users, environment, organization</td>
</tr>
<tr>
<td>Virus attack</td>
<td>EP</td>
<td>System goal/objective, infrastructure</td>
<td>June to Sep. 1995</td>
<td>Confusion and frustration</td>
<td>Users, developer, organization, environment</td>
</tr>
<tr>
<td>Threat to Quit</td>
<td>ST</td>
<td>All levels</td>
<td>August 20, 1995</td>
<td>Desperation and frustration</td>
<td>Users, developer, organization, environment</td>
</tr>
</tbody>
</table>

4.12. Recovery and Slow Growth Toward Early Implementation

A series of processes revived the project during this next phase of development. By mid-September 1995, the project was stalled. Once again, however, the developer’s threat to

171 Political maneuvers and power relationships threatened the project’s success (Markus 1983). Key resources to the project’s continuation were being lost. The clinic was not achieving self sufficiency (Attewell 1992). Project escalation resulted in a need to cut losses (Keil 1994).
leave revived the clinic’s support for the project, and coincided with a number of other fortunate events (Markus 1981; Orlikowski & Robey 1991).

The Monkey II virus was identified as the cause of the computer crashes, and calmed the fears and hostility about OS/2 by providing the research group with an identifiable cause and solution (Gibbs 1994).

The research group also took the lead on the project, as the developer moved to the background as a consultant and software advisor. This leadership has been important in achieving the viability of SoftHeart away from its developer (Kidder 1981; Basalla 1988; Attewell 1992; Petroski 1994). The purchase of computers and network hardware and software started to produce visible signs of internal resource support, and coincided with the research group’s movement into the lead (Kling & Scacchi 1982; Earl 1989; King et al., 1994). Additional resources arrived, including funding from a drug company, and the hiring of a programmer.

---

172 The loss of a “free” developer and someone who knew the SoftHeart software caused a shakeup in the organization (Markus 1981). A key resource for future research capabilities and clinical care was threatened by the developer’s dissatisfaction (Orlikowski & Robey 1991).

173 Technical problems were identified and new safety strategies were constructed (Gibbs 1994).

174 The Monkey II virus is a boot sector virus released in early 1994, that infects the boot sector of a disk. The virus spreads by activation of a floppy disk in the floppy drive, and copies itself onto the hard drive of a computer. This virus then operates in memory, affecting other fresh floppy disks inserted into the computer. The virus eventually destroys the boot sector of the hard drive making it impossible to access the drive. Protection from the virus required anti-virus software, and “safe” computer practices.


176 SoftHeart’s “web” was being developed to support the IT infrastructure (Kling & Scacchi 1982). Strategic acquisition of resources occurred (Earl 1989). Institutional support and outside subsidization of the project appeared (King et al. 1994).
Early implementation has been difficult because of limited time with busy users, learning \(^{177}\) (Argyris, Putnam, & Smith 1985; Van De Ven & Polley 1992; Attewell 1992), mismatches between technology and task \(^{178}\) (Cooper & Zmud 1990), misunderstanding and early conflict \(^{179}\) (Markus 1983), and technical bugs that created user concerns and weakened confidence \(^{180}\) (Henderson & Kyng 1991; Wynn 1991).

The following summarizes the processes in this stage:

**Table 4-12: Processes in the Recovery Stage**

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for the project revives</td>
<td>ST</td>
<td>Infrastructure</td>
<td>August to Sep. 1995</td>
<td>Optimism</td>
<td>Developer, users, directors</td>
</tr>
<tr>
<td>Research group takes the lead</td>
<td>EP</td>
<td>System Objectives</td>
<td>Sep. to Dec. 1995</td>
<td>Project begins to live by itself</td>
<td>Research group</td>
</tr>
<tr>
<td>Resources arrive</td>
<td>EP/OI</td>
<td>Infrastructure, program</td>
<td>Sep. 1995 through Jan 1996</td>
<td>Impetus</td>
<td>Funders, computer suppliers, research group, developer</td>
</tr>
<tr>
<td>Early Implementation</td>
<td>EP/OI</td>
<td>Program, input, display, reporting, data management, data</td>
<td>Feb. to April 1996</td>
<td>Concern, Chaos, and Cooperation</td>
<td>Secretaries and Clerks, developers/ suppliers</td>
</tr>
</tbody>
</table>

\(^{177}\) Problems with single loop learning and a struggle to drive double loop learning began (Argyris, Putnam & Smith 1985; Van De Ven & Polley 1992). Learning and the knowledge enhancing or destroying aspects of technology were highlighted (Attewell 1992).

\(^{178}\) A mismatch between technology and task in this case was similar to the task-technology literature (Cooper & Zmud 1990).

\(^{179}\) Misunderstandings between administrative and development staff appeared (Markus 1983).

\(^{180}\) There was a weakening of confidence by producing a software that did not fully consider clerical work (Wynn 1991) and lacked user testing and tailoring (Henderson & Kyng 1991).
4.13. What Lies Ahead

The following provides a number of processes outlining the future of SoftHeart. Structural power both threatens and supports the operation and the impact of SoftHeart (Dutton 1981; Markus 1981; Thomas 1994). The ability of the organizational and technical environment to support SoftHeart’s operation and research focus is fairly strong (Kling & Scacchi 1982; Earl 1989). However, the support of SoftHeart’s original goal toward patient care and the direct impact on improving patient care is uncertain. The growing role of LipidLink and the research group has shifted the project’s focus from patient clinical care to research (Dutton 1981; Markus 1981; Van De Ven 1986). The following summarizes these processes:

1 Politics within the clinics, the hospital and the government both threatened and introduced opportunities for SoftHeart to play a role (Markus 1981). Structural power and interests in the clinics moving SoftHeart away from patient clinical care to research (Dutton 1981; Markus 1981; Thomas 1994).

2 Resources available to support the continued development and infrastructure development appeared (Kling & Scacchi 1982). Strategic benefits of SoftHeart appearing with competition for government and hospital funding (Earl 1989).

3 The changed development environment around SoftHeart has altered its role from the developer’s focus on patient and clinical care, to the research group’s focus on a database for research. Dutton (1981) discusses the effect of a changed development environment on technology adaptation and rejection. Markus (1981) discusses how technology will only survive if it is adapted to the power relationships in the context. Van De Ven (1986) discusses the need for a core concept, and the focusing of resources on that concept to develop an innovation. In this case, SoftHeart is being stretched in a number of competing directions, and has created a recent funding crisis.

4 Patient clinical care is becoming the least important focus of implementation and development efforts. Instead, research and the administrative data entry and personnel responsible for data entry, have been the focus.
Table 4-13: Processes in the Future of SoftHeart

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threats to Operation</td>
<td>ST</td>
<td>Project conception</td>
<td>Future</td>
<td>Uncertainty</td>
<td>Developer, research group, MIS department, drug company</td>
</tr>
<tr>
<td>Threats to Impact</td>
<td>ST/OI</td>
<td>Project conception</td>
<td>Future</td>
<td>Uncertainty</td>
<td>Developer, research group, clinics, directors</td>
</tr>
</tbody>
</table>

4.14. Case Conclusion

An analysis of the processes, process stages and the overall ISD process will provide some insights into the four process theories.


In terms of *mutual exclusivity*, the following table shows the process types in each process stage of SoftHeart’s development:


Table 4-14: Summary of Process Types in Process Stages

<table>
<thead>
<tr>
<th>Process Stage</th>
<th>Number of Processes</th>
<th>Process Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Emergence</td>
<td>3</td>
<td>EP, EP, OI/TI</td>
</tr>
<tr>
<td>Health Funders Skeptical</td>
<td>2</td>
<td>EP/TI, EP/ST</td>
</tr>
<tr>
<td>Our Response</td>
<td>3</td>
<td>ST/EP, OI, OI/EP</td>
</tr>
<tr>
<td>Presentation Sparks</td>
<td>3</td>
<td>EP, EP, EP/ST</td>
</tr>
<tr>
<td>Interest and Constraint</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conflicts Erupts</td>
<td>4</td>
<td>EP/ST, ST/OI, OI/TI, EP</td>
</tr>
<tr>
<td>Project Builds Momentum</td>
<td>4</td>
<td>ST/EP, OI/EP, OI/ST, ST</td>
</tr>
<tr>
<td>Future</td>
<td>2</td>
<td>ST, ST/OI</td>
</tr>
<tr>
<td>ISD Overall</td>
<td>43</td>
<td>TI 3, 3 Shared, OI 15, 13 Shared, EP 31, 15 Shared, ST 16, 12 Shared</td>
</tr>
</tbody>
</table>

The following table shows the ISD research attached to the process stages (see table 2-6):
<table>
<thead>
<tr>
<th>Process Stage</th>
<th>Process Types</th>
<th>ISD Research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Emergence</td>
<td>TI (1)</td>
<td>Technology; System</td>
</tr>
<tr>
<td></td>
<td>OI (1)</td>
<td>Task technology fit;</td>
</tr>
<tr>
<td></td>
<td>EP (2)</td>
<td>Designer’s view; Evolution of technology; Innovation</td>
</tr>
<tr>
<td>Health Funders Skeptical</td>
<td>TI (1)</td>
<td>Technology; Systems</td>
</tr>
<tr>
<td></td>
<td>EP (2)</td>
<td>Evolution of technology; User involvement; Innovation</td>
</tr>
<tr>
<td></td>
<td>ST (1)</td>
<td>Adaptive structuration</td>
</tr>
<tr>
<td>Our Response</td>
<td>OI (2)</td>
<td>Structured information systems planning; Structured design and analysis; User involvement</td>
</tr>
<tr>
<td></td>
<td>EP (2)</td>
<td>Evolution of technology; User involvement; Innovation</td>
</tr>
<tr>
<td></td>
<td>ST (1)</td>
<td>Adaptive structuration; Structural power</td>
</tr>
<tr>
<td>Old and New Environmental Shocks</td>
<td>EP (3)</td>
<td>Technology divergence (Uni); Predicament of design; Evolution of technology; Innovation</td>
</tr>
<tr>
<td></td>
<td>ST (2)</td>
<td>Adaptive structuration; Structural power</td>
</tr>
<tr>
<td>Decision to Move Ahead</td>
<td>OI (5)</td>
<td>Strategic information systems planning; User involvement; Task technology fit</td>
</tr>
<tr>
<td></td>
<td>EP (4)</td>
<td>Predicament of design; Emergent evolution of technology; Innovation</td>
</tr>
<tr>
<td></td>
<td>ST (2)</td>
<td>Adaptive structuration; Structural power</td>
</tr>
<tr>
<td>Prototype Presentation Sparks Interest and Constraint</td>
<td>EP (3)</td>
<td>Developer’s view; Alternative views; Predicament of design; User involvement; Innovation</td>
</tr>
<tr>
<td></td>
<td>ST (1)</td>
<td>Adaptive structuration; Structural power</td>
</tr>
<tr>
<td>Conflict Erupts</td>
<td>TI (1)</td>
<td>System</td>
</tr>
<tr>
<td></td>
<td>OI (2)</td>
<td>Task technology fit</td>
</tr>
<tr>
<td></td>
<td>EP (2)</td>
<td>User involvement</td>
</tr>
<tr>
<td></td>
<td>ST (2)</td>
<td>Adaptive structuration; Structural power</td>
</tr>
<tr>
<td>Decision to act</td>
<td>EP (4)</td>
<td>Technology divergence (Uni); User involvement; Innovation</td>
</tr>
<tr>
<td>Project Builds Momentum</td>
<td>OI (2)</td>
<td>Strategic information systems planning</td>
</tr>
<tr>
<td></td>
<td>EP (2)</td>
<td>Technology divergence (Uni); Predicament of design; User involvement</td>
</tr>
<tr>
<td></td>
<td>ST (3)</td>
<td>Adaptive structuration; Structural power</td>
</tr>
</tbody>
</table>
Chapter 4: Processes in the SoftHeart Case

<table>
<thead>
<tr>
<th>Development Quagmire</th>
<th>EP (3)</th>
<th>Technology divergence; Predicament of design; User involvement; Innovation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ST (1)</td>
<td>Structural power</td>
</tr>
<tr>
<td>Recovery and Slow Growth to Early Implementation</td>
<td>OI (2)</td>
<td>Strategic information systems planning</td>
</tr>
<tr>
<td></td>
<td>EP (6)</td>
<td>Technology divergence (Uni); Predicament of design; Evolution of technology; User involvement; Innovation</td>
</tr>
<tr>
<td></td>
<td>ST (3)</td>
<td>Structural power</td>
</tr>
<tr>
<td>Future</td>
<td>OI (1)</td>
<td>Strategic information systems planning</td>
</tr>
<tr>
<td></td>
<td>ST (2)</td>
<td>Adaptive structuration; Structural power</td>
</tr>
</tbody>
</table>

Keeping in mind the subjective selection of critical events and processes, the following observations can be stated. First, the emergent perspective is linked with thirty one of forty three processes, a result of the complex environment at the two clinics and the role of the developer. Emergence affected all levels of technology, but substantially affected lower levels such as interface design. Emergence best described several important processes including project initiation, external shocks from health funders, new stakeholder groups joining the project, the developer’s influence and learning on design, prototyping, internal shocks from political activity inside the clinics, and historical conflict between various groups.

Second, the social technology perspective and organizational imperative are used to describe sixteen and fifteen processes respectively, but a large number of these processes (twelve and thirteen respectively) are also described using other theories. The technological imperative was used to describe the fewest of the case processes, and describes none of the processes on its own. This may reflect the early development period, and the rapid changes in ideas and software during this stage. Perhaps later stages of implementation would be explained more by a technological imperative once the software is “closed”.
Examining *special case* and *equal validity* explanations, the following table summarizes the "shared" processes:

### Table 4-16: Examination of Shared Processes

<table>
<thead>
<tr>
<th>Process</th>
<th>Total Shared*</th>
<th>with TI</th>
<th>with OI</th>
<th>with EP</th>
<th>with ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>3</td>
<td>-</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>OI</td>
<td>13</td>
<td>2</td>
<td>-</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>EP</td>
<td>15</td>
<td>1</td>
<td>7</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>ST</td>
<td>12</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>-</td>
</tr>
</tbody>
</table>

* Note: the "total shared" numbers are one smaller than the sum in the OI, EP and ST rows because of a triple-coded process.

The table shows a connection between the emergent perspective and organizational imperative (seven shared), emergent perspective and social technology (eight shared), and social technology and organizational imperative (five shared). Each may be special cases of the other. For example, the connection between the organizational imperative and the other two implies that strategic considerations occur within an environment and set of technological constraints or opportunities. The link between emergence and social technology may indicate a fine line between social technology's soft-line determinism and the unpredictable nature of emergence.\(^{185}\)

Examining *association* between process types over time, the following table summarizes the number of times a process type followed another process type:

---

\(^{185}\) In fact Barley (1986), an adaptive structuration and social technology researcher, states that many explanations of structure's effect on technology and vice versa can only be explained through historical analysis.
Table 4-17: Number of Times One Process Type is Followed Another Type

<table>
<thead>
<tr>
<th>Process</th>
<th>Total</th>
<th>by TI</th>
<th>by OI</th>
<th>by EP</th>
<th>by ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>OI</td>
<td>21</td>
<td>2</td>
<td>7</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>EP</td>
<td>47</td>
<td>1</td>
<td>11</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td>ST</td>
<td>22</td>
<td>1</td>
<td>5</td>
<td>9</td>
<td>7</td>
</tr>
</tbody>
</table>

Once emergence appeared, it was usually followed by another emergent process (twenty three of forty seven). Combining social technology and organizational imperatives described another twenty three of the forty seven emergent “followings”, suggesting a close relationship among the theories. This close relationship among the emergent perspective, social technology and the organizational imperative was illustrated in the other numbers. An organizational imperative was followed by an emergent process in eight of twenty one, an organizational imperative in seven of twenty one, and a social technology process in four of twenty one. In addition, social technology was followed by another social technology process in seven of twenty two, and an emergent process nine of twenty two times. Finally, technological imperative “followings” occurred only five times, but three of these five were followed by an emergent perspective, suggesting a connection between the two.

The next table shows the likelihood of a series of processes with one type being terminated by one or more other process types, and provides evidence of a possible connection between the organizational imperative and the emergent perspective:
Table 4-18: Number of Times a Series of One Process Type was Terminated by Another Type

<table>
<thead>
<tr>
<th>Process</th>
<th>Total</th>
<th>by TI</th>
<th>by OI</th>
<th>by EP</th>
<th>by ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>3</td>
<td>-</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>OI</td>
<td>8</td>
<td>1</td>
<td>-</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>EP</td>
<td>10</td>
<td>1</td>
<td>5</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>ST</td>
<td>9</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>-</td>
</tr>
</tbody>
</table>

A series of processes described by the emergent perspective were terminated by processes described as organizational imperatives and social technology on five and four respective occasions. On the other hand, a series of processes described by organizational imperative were usually terminated only by an emergent process (five of eight), while a series of social technology processes are terminated by organizational imperatives and emergent perspectives in three and five times respectively. This finding implies that organizational imperatives may be “disturbed” by new emergent processes, and a series of social technology or emergent perspectives may be “disturbed” equally by the other two process types.

### 4.14.2. Information System Development

The predominance of emergent processes suggests an interplay between the emergent perspective and the other process theories, especially social technology, over time. As a result, many process stages in the case were emergent, and ISD was overall emergent. Adaptation to sudden shocks or learning sent the project into reflective disarray. To achieve control once again, it appears that social technological constraints or opportunities were incorporated into the project’s concept, the project direction was modified, and development continued. Occasionally, the developers were able to alleviate many of the constraints to create an organizational imperative process.
5. Processes in the Initiation and Development of EMPOWER

The development of EMPOWER involved a series of processes between technology and context over time. Important in these processes were various stakeholders including the developer (myself), the project leader, the development team, funding agencies, the US company, health promotion researchers, and software suppliers.

This chapter reports on critical and illustrative technology-context processes within “process stages” in the development of EMPOWER. Processes, process stages, and the ISD are discussed using the four process theories and research of chapter 2. Citations to the literature and inference rules for each process are provided.

First, the chapter describes the development team, the vision of the EMPOWER’s role in breast cancer planning, and the software in its present form. The majority of the chapter examines the various processes appearing in the process stages of EMPOWER’s development. The following table summarizes these process stages:

---

186 At the time of writing this thesis, EMPOWER had reached the early implementation stages in two communities, and development of a new EMPOWER had recently begun.

187 Appendix B provides a more detailed account of the chronological events in the EMPOWER case, and provides the raw data used to build this chapter.

188 Note that these stages, although presented sequentially, were often overlapping.
Table 5-1: Process Stages in the Development of EMPOWER

<table>
<thead>
<tr>
<th>Process Stage</th>
<th>Time Period</th>
<th>Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Emergence</td>
<td>November 1993 - April 1995</td>
<td>Initiation</td>
</tr>
<tr>
<td>Getting Ready</td>
<td>March - September 1995</td>
<td>Adoption</td>
</tr>
<tr>
<td>Pandora’s Box Opens</td>
<td>September - October 1993</td>
<td>Adoption and Adaptation</td>
</tr>
<tr>
<td>Squeezing the Most from Toolbook and EMPOWER</td>
<td>September 1995 - April 1996</td>
<td>Adoption and Adaptation</td>
</tr>
<tr>
<td>Visions of a New Software Constrained by Resources</td>
<td>December 1995 - April 1996</td>
<td>Initiation, Adoption, Adaptation and Acceptance</td>
</tr>
</tbody>
</table>

The chapter concludes with a discussion of the case and the four process theories’ explanation of EMPOWER’s development.

5.1. The Development Team and EMPOWER

The development of a Canadian and international version of EMPOWER was carried out by a team of health promotion researchers, including the developer. The purpose of development was to produce a Canadian and international version of EMPOWER from the US version of the software, and to implement and observe the usage of the software in two Canadian communities planning mammography screening interventions. Further adaptation of the software would be completed from experiences in the field.

The team consisted of the principal investigator and project leader, a co-investigator for the BC case study, a co-investigator for the Quebec case study, research assistants at the University of British Columbia and the University of Montreal, a developer (me), a programmer, and one secretary. Other individuals were hired and/or volunteered for various parts of the project. The project was funded by the National Cancer Institute of Canada (NCIC), which was interested only in the implementation and development of a planning
methodology across Canadian communities. NCIC provided a 2 year grant for the pilot testing of the planning tool in the Canadian settings.

EMPOWER (Expert Methods of Planning and Organizing Within Everyone's Reach) is a Windows standalone software program to support planners of mammography screening interventions in community settings. The planning model represented in the software is based on the project leader's model of health promotion planning, which has been employed successfully in numerous settings. The model employs a bottom-up (participatory) and top-down (scientific and technical) approach to planning, assessing the resources of the community for planning (situational analysis), the needs of the population (social analysis), the health issues in the population affecting these needs (epidemiological analysis), the behavioral and environmental factors affecting health (behavioral and environmental analysis), the predisposing, reinforcing, and enabling factors affecting those behaviors and environment (educational and organizational analysis), and the educational, policy, regulatory, and organizational factors needed to change those factors (administrative and policy analysis).

In addition, "consult-on-tap" resources are available, including health promotion model references, a combined health information database, case examples, "responding here" help, page references, university and non-university resources, clinical oncology centers, the American 2000 health objectives, intervention ideas, and glossaries.

The software platform used to build EMPOWER is Asymetrix's Multimedia Toolbook. Based on the metaphor of a book, the programmer creates books with pages,
backgrounds, and fields for navigating pages, reading information, and entering data. Multimedia capabilities are used in EMPOWER to catch user attention, and to provide information about the planning stage described on the page. In addition, menus provide access to “consult-on-tap” resources.\textsuperscript{189}

EMPOWER consists of 6 basic books to address the six stages of the planning model, and is linked together to provide the users with a sequential view of the planning model and their project. Information about the specific project is entered into a project file, and this information is then used in “expert” summary reports that summarize the data, provide some tips for planning, and display scientific evidence or professional experience in dealing with circumstances described (categorized) by the user.

The program structure is displayed below:

\textsuperscript{189} See Appendix F to view some of the screens in EMPOWER.
Currently, EMPOWER is being used in two Canadian communities to assist in developing a mammography screening campaign to increase women’s involvement in mammography screening. Initial results confirm that the users use the software for initial planning and as a resource. In the first community (Richmond, B.C.), however, the software is not being used as a planning tool to organize and produce the plan continuously throughout the process. The second setting (Montreal) is just beginning its use of the software at the time of this writing.

The following sections provide a stage-by-stage description of the ISD process of the Canadian and international version of EMPOWER.
5.2. Project Emergence

Important in understanding information systems development are the processes resulting in the initiation and creation of IS and IT. The EMPOWER project emerged from a series of opportunities and interests between the developer, the director, the project team members, the funding agencies, and the US company stakeholders (Basalla 1988; Petroski 1994; Thomas 1994). The opportunities perceived in the project varied amongst the individual members of the team. For the project leader, EMPOWER represented an opportunity to diffuse his model of planning to many users, and through redevelopment, to free the software from the exclusive control of the US company. The project also provided the project leader with an opportunity to redevelop EMPOWER for other health areas, and to produce a generic version to reach many users. For the developer, the project was an opportunity to complete his thesis with a project that would lead to software use. It also provided an opportunity to form closer ties with health promotion researchers. For other team members, the project represented a chance to be a part of an important research grant, to work with the project leader, to work with leading edge technology, to develop a tool to improve planning, and to put their fingerprints and ideas on a potentially wide-reaching tool.

For the communities involved in testing the software, the project represented an opportunity to build better plans.

---

190 The group identified an opportunity to improve and modify the US version of the software for Canada (Basalla 1988). The evolution of a technology and emergence occurred through suppliers (Petroski 1994; Thomas 1994).

191 Negotiations between the project leader and the US company over development, use, copyright and marketing of subsequent versions of the software was the developer's first introduction to the project (January 1994). The developer concluded that because the project leader was paid by the US company, the US version of the program was property of the US company. Any modification of the program would require a percentage royalty to the US company based on the percentage of the original code in the product.
During 1994, meetings about the US copyright and a discussion with Statistics Canada about downloading epidemiological data for EMPOWER, demonstrated the “web of computing” around the project (Kling & Scacchi 1982). In addition, the developer’s increasing involvement with a number of software projects with the health researchers fused his interests in studying software development in health settings to the health promotion group’s interest in using software as a medium for changing health policy and planning behavior (Earl 1989; Hirschheim & Klein 1989; Attewell 1992; Thomas 1994).

During early 1995, project directions, constraints, commitments, opportunities, and resources arose from discussion and external influences (Dutton 1981; Kling & Scacchi 1982; Markus 1983; Orlikowski 1991; Thomas 1994). The problem definition and problem solution were generated and changed simultaneously based on the funding agency that would eventually support the project and the numerous goals and issues raised (Dutton 1981; Kling & Scacchi 1982; Turner 1987; Thomas 1994). Originally focusing on tobacco control, it was soon realized that heart disease, mammography screening, and/or self care were other potential areas of redevelopment depending on funding agency responses. When a research

192 Web influences around the project included the history of the US version and future connections with data suppliers (Kling & Scacchi 1982).

193 The software represented a medium for changing health policy and planning (Earl 1989). There was an integration of interests and goals toward software development (Hirschheim & Klein 1989). EMPOWER involved learning and supply side innovation (Attewell 1992; Thomas 1994).

194 There was a contextual assessment of commitments, opportunities, constraints and assumptions of development (Dutton 1981). This analysis represented a web analysis of key players and goals (Kling & Scacchi 1982). Structural power and different interest groups were identified (Markus 1983). Pressures and forces shaping design were identified (Orlikowski 1991). Process-political considerations of EMPOWER’s development were discussed (Thomas 1994).

195 The setting of development was being created at the same time as the software ideas were being formulated (Dutton 1981). Resources to support development and implementation of the software in two communities were being considered (Kling & Scacchi 1982). Problem generation and solutions considered simultaneously (Turner 1987). Supplier led development initiated by developers’ visions (Thomas 1994).
grant proposal for tobacco control was not funded, a three week period of confusion appeared\(^{196}\) (Dutton 1981; Kling & Scacchi 1982). When the National Cancer Institute of Canada decided to fund a separate grant proposal that had been submitted earlier, the project shifted focus from the redevelopment of EMPOWER toward tobacco control, to the modification and field testing of the US mammography screening version in Canada. The pieces fell into place\(^{197}\) (Earl 1989; Orlikowski & Robey 1991).

The following table outlines the important processes in the emergence stage:

---

\(^{196}\) Loss of original context made original plans irrelevant (Dutton 1981). Web resources for funding and target audience failed (Kling & Scacchi 1982).

\(^{197}\) There was a strategic direction and leveraging of existing US version focused on mammography (Earl 1989). Institutionalization and context affects the constraints, opportunities and governing principles of the project (Orlikowski & Robey 1991).
Table 5-2: Processes in the Emergence Stage

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opportunity</td>
<td>EP</td>
<td>Project conception</td>
<td>Nov. 1993 - Jan 1996</td>
<td>Exploration and discussion</td>
<td>Developer, project leader</td>
</tr>
<tr>
<td>Increasing involvement with health promotion group</td>
<td>EP/OI</td>
<td>Project conception</td>
<td>June 1994 - Jan 1995</td>
<td>Interest and exploration</td>
<td>Developer, project leader, other health promotion groups</td>
</tr>
<tr>
<td>Meetings of Tobacco control project</td>
<td>EP/ST</td>
<td>System objectives and system functionality</td>
<td>Feb. - Mar. 1995</td>
<td>Organization</td>
<td>Developer, project leader, US company, UBC project team</td>
</tr>
<tr>
<td>Tobacco grant rejected</td>
<td>EP</td>
<td>System objectives</td>
<td>April 1995</td>
<td>Surprise and loss of momentum</td>
<td>Health funders, UBC project team, Health Canada</td>
</tr>
<tr>
<td>Mammmography screening funded</td>
<td>OI/ST</td>
<td>System objectives</td>
<td>April 1995</td>
<td>Rejuvenation and reorientation</td>
<td>Developer, project leader, NCIC funder, UBC project team, US company</td>
</tr>
</tbody>
</table>

5.3. Getting Ready

During the Getting Ready stage, the development team focused on the goals and constraints of modifying the EMPOWER software for the Canadian and international setting. Many visions, ideas and the uncertainty about the US software created a large number of
complex paths to consider \(^{198}\) (Nadler 1982; Kling 1987; Turner 1987; Orlikowski 1991). Choices were made with uncertain information about their positive and negative impacts on the project.

To cope with the complexity, one of the early structural decisions was the separation of the “content” group from the “structural” (programming) group. The structural group was to be led by the project leader with the developer as the programming supervisor. The content group was to be led by the co-investigator on the project team.\(^ {199}\) To form a common language, the team used the Toolbook language (books, pages, buttons, etc.), providing a common ground for discussion between the content and structural groups \(^ {200}\) (Van De Ven 1986; Boland 1990).

The most important decision facing the project at that point was the “make” versus “buy” decision; whether to adopt the US version and modify it for the Canadian setting, or whether to start from scratch. Advantages of modifying the US version appeared to include a nearly completed product and beta-tested source code thereby reducing the need for extensive user involvement, a proven tool for expert planning \(^ {201}\) (Sviokla 1989; DeLone & McLean 1992), a good template for future software inventions \(^ {202}\) (Petroski 1994), fewer resources

---

\(^{198}\) Many visions and ideas about the course of development is similar to the problems of complexity outlined by Nadler (1982). Setting software development boundaries was an important but overlooked issue during early stages (Kling 1987). Problem definition and solution simultaneously considered (Turner 1987). The planning process was to be tightly embedded in the software technology. The number of matches between planning and tools for planning infinite (Orlikowski 1991).

\(^{199}\) Without direct user involvement, the content group represented the “users” of the software.

\(^{200}\) Whole part management of the project important (Van De Ven 1986). Hermeneutics and team members’ background affected the goals, discussion of the goals and task specialization (Boland 1990).

\(^{201}\) Use of expert systems, and evaluating its impact important in the project (Sviokla 1989). Task technology fit assessed (DeLone & McLean 1992).

\(^{202}\) Evolution or revolution of the US version was discussed. Obvious shortfalls in the US version were uncovered (Petroski 1994).
required relative to developing a Canadian version from scratch, and backup support from the
US company 203 (Earl 1989). 204 Advantages of starting from scratch included: freeing
ourselves from the US company over access to the code documentation and copyright,
avoiding the difficulty of learning the Toolbook programming language and deciphering the
existing code, building a faster program, decreasing the size of the program 205, addressing the
fragmented storage of information, and overcoming some undesirable interface characteristics
of the US version 206 (Markus 1981; Kling & Scacchi 1982). 207 A third strategy included a
number of intermediate approaches between “buy” and “make”, including outsourcing of
functions to other programs, and handling most database intensive functions in a separate
database platform that would be called by Toolbook.

Important in these considerations was the exploration of the US software package to
evaluate its technical and functional capabilities. Various factors inadvertently favored the
“buy” decision such as limited time, a course with the Toolbook company that favorably
influenced the developer’s impressions of the software, and the purchase of the Database
Connection marketed by the Toolbook company to potentially solve data storage problems in

---

203 Strategic resource considerations were evaluated (Earl 1989).
204 During June of 1995, we did receive some indication that the package had only been tested in educational
   environments, as part of its demonstration in workshops. We also received warnings about modifying the code from the
   leader of the US team.
205 Currently, EMPOWER requires over 12 Megabytes of disk space.
206 Previous commitments and copyright restrictions with the US company considered (Markus 1981). The web around
development was in formation and evaluation (Kling & Scacchi 1982).
207 These poor interface characteristics included page-by-page navigation, lack of standardized screen appearance, and the
   large number of pages (96).
the US version \( \text{\textsuperscript{208}} \) (Kling & Scacchi 1982; Attewell 1992; Tyre & Orlikowski 1994). As a result, the developer recommended the project team redevelop the US version in late August.

With the buy decision behind us, the remaining resources fell into place during early September, including the hiring of a programmer whose influence on the project would be important \( \text{\textsuperscript{209}} \) (Kidder 1981; Earl 1989). The following table summarizes the processes involved in this stage:

**Table 5-3: Processes in the “Getting Ready” Stage**

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Division of labor</td>
<td>EP</td>
<td>Program, data structure</td>
<td>Mar. - Aug. 1995</td>
<td>Specialization based on expertise</td>
<td>Developer, and health promotion researchers</td>
</tr>
<tr>
<td>Addition resources fell into place</td>
<td>ST/OI</td>
<td>Infrastructure</td>
<td>Sept. 1995</td>
<td>Acquisition</td>
<td>Developer, project leader, UBC project team, NCIC funder.</td>
</tr>
</tbody>
</table>

\( \text{\textsuperscript{208}} \) The formation of the web of resources began to support development by purchasing the database connection to solve data storage problems (Kling & Scacchi 1982). Learning and barriers to implementation considered (Attewell 1992). Drive for production and time constraints limited the learning and web formation time (Tyre & Orlikowski 1994).

\( \text{\textsuperscript{209}} \) Team members’ personalities and expertise important for success (Kidder 1981). Strategic resource assessment and procurement completed (Earl 1989).
5.4. *Pandora's Box Opens*

Almost simultaneously as the group was preparing for development, problems were uncovered in the US version of EMPOWER. Examination of the US version revealed some potentially serious problems. Blinded by the possibility of quick success, however, momentum prevented us from responding actively to this evidence *(Van de Ven & Polley 1992).*

Without documentation, the programmer and the developer were faced with the difficulty of exploring code embedded in numerous objects on each page of the US version of EMPOWER, including push buttons, fields, pages, backgrounds, books, and four large system books. This code was very frequently undocumented except for brief comments, and a significant portion of this code is still unknown and potentially old code that is no longer being used. The problems with object orientation also became an issue with case examples, ‘responding here’ help information for each page, and references which to our surprise were properties of each page. This made it extremely difficult to review and change the information *(Brooks 1987; Turner 1987; Gibbs 1994).* As a result of unclear documentation, changes to the program often had unexpected and unwanted results. On many occasions, a single change turned into a string of changes required to fix rippling software bugs *(Nadler 1982; Brooks 1987; Turner 1987; Harrington 1991).*

---

210 A clear case of an inability to learn from the symptoms of design flaws in the US version as the lure of quick success prevented double-loop learning *(Van De Ven & Polley 1992).*

211 Invisibility and complexity of design caused changes to ripple *(Brooks 1987).* Complex and abstract software with poor documentation hindered learning *(Turner 1987).* Poor initial design in US version became apparent *(Gibbs 1994).*

212 The complex environment around design and complex US design caused confusion and anxiety *(Nadler 1982).* The invisibility and complexity of software made it difficult to analyze and change *(Brooks 1987; Turner 1987).* Change to software caused unexpected results in program operation and user interface *(Harrington 1991).*
Discussion with the project leader and the two colleagues identified a number of reasons for the software's complexity. First, the scope of the project extended beyond the resources available at that time (Van De Ven 1986). Second, the US company's organizational approach to late-stage development was to farm out the software to a private developer who had previously been on the in-house programming staff. This eliminated the chance for closer communication between the US and Canadian programmers and monitoring of activities (Van De Ven 1986; Keil 1995). Third, both educational and planning objectives were incompletely built into the software by the two colleagues holding these views. This increased EMPOWER's size and produced a mixture of pure educational and pure planning pages (Kling 1980; Kling & Scacchi 1982; Van De Ven 1986; Boland 1990; Petroski 1994).

In addition, problems with “content” in the software indicated that the program might no longer be relevant to the users' setting (Dutton 1981). First, changes in research approaches toward full breast health, including breast self examination (BSE) and clinical breast examinations (CBE), may have made the singular focus on mammography screening demands exceeding the resources available, similar to Van De Ven’s four central problems of innovation (Van De Ven 1986).

14 Inability to learn the US programmer's approach to design caused a reduction in whole-part coordination (Van De Ven 1986). The result was the basis of project escalation and increased commitment (Keil 1995).

15 A mixture of planning and educational suggested a possible confusion in approaches and overall goals (Kling 1980; Kling & Scacchi 1982). Reasons for the software poor design included demands exceeding initial resources, and loss of key programmer that sealed the product to easy modification (Van De Ven 1986). There was a hermeneutic confusion trying to understand a foreign object (Boland 1990). The old software was critiqued as evolution begins (Petroski 1994). The loss of key programmer on the project resulted in loss of an essential resource (Keil 1995).

16 In fact, some modules were designed primarily by the educational proponent (e.g. situational analysis), and some modules primarily by the planning proponent (e.g. administrative and policy analysis). Middle modules were handled by the programming staff.

17 Dutton's (1981) study of a change in local governments and their policies effect on the development and relevance of a land planning system provided insights into this EMPOWER process.
unpalatable to researchers and planners in Canada. Second, new references and the rewriting of poorly written case examples were also required to make the software relevant. Finally, four different versions of EMPOWER 1.1 were discovered, illustrating symptoms of project management problems at the US company \(^{218}\) (Van De Ven 1986; Latour 1987). Each one of the four versions contained incomplete information, and each version had more complete data in certain areas. This required a building of an integrated version that included the correct data from all four versions.

One of the fallback scenarios the UBC team had agreed upon as a contingency plan, if the EMPOWER software proved difficult to change, was the outsourcing of important functions to other software. However, attaching dynamic link libraries (DLL) to Toolbook was very difficult, and the database connection failed to produce a stable and reliable connection with the files \(^{219}\) (Kling & Scacchi 1982; Brooks 1987; Turner 1987). In addition, the use of other database packages to access the program files to update content caused unexpected structural changes to the files. The team found itself stuck with a software “island”, and with little time left until software implementation in community one \(^{220}\) (Orlikowski & Robey 1991).

---

\(^{218}\) Problems with project management at the US company affected and confused the development team (Van De Ven 1986). The development team opened the black box and discovered problems and controversy embedded in poor software code (Latour 1987).

\(^{219}\) The web of computing components failed to fix data management problems (Kling & Scacchi 1982). Complexity and time was consumed working with abstract and poorly documented Dynamic Link Libraries (DLL) (Brooks 1987). The complexity of design and poor support hampered efforts (Turner 1987).

\(^{220}\) The software failed to meet development team expectations, suggesting that structural goals were constraining and directing development (Orlikowski & Robey 1991).
Meanwhile, the development list continued to grow beyond "Canadianization" of the software. Many of the software bugs required detailed work. In addition, many of the development expectations generated by early enthusiasm caused development requirements to expand \(^{221}\) (Nadler 1982; Van De Ven 1986). Large lists of changes by numerous individuals were compiled in early September and continued to grow throughout the four months. Ideas and visions of international and generic versions of EMPOWER, that would serve many disease and health promotion areas, also increased the "wish" list. The team lost sight of a core concept. The change lists reflected a reactive approach to the software, where problems and functions on each of the ninety six pages drove development \(^{222}\) (Van De Ven 1986). Being rather loosely constructed, the US version led us in many directions fixing functions that in retrospect may not be essential to health promotion planning \(^{223}\) (Van De Ven 1986; Brooks 1987; Mohr 1987).

The developer's reaction to the development problems and the US version were mixed. At various points throughout the project, the developer suggested switching to an alternative software platform, and interjected that the US software was poorly designed. The result would be a product that would fail as a planning tool. At other times, he stated that the

---

\(^{221}\) Demands and promises from various individuals and groups, combined with inability to penetrate the US version of the software, increased complexity (Nadler 1982). Project requirements and programming team abilities were diverging (Van De Ven 1986).

\(^{222}\) Software drove the core concept and obscured a core vision (Van De Ven 1986).

\(^{223}\) Management of the project spiraled out of control (Van De Ven 1986). Changeability and expansion in social demands from various groups increased complexity (Brooks 1987). The malleable and expansive nature of IT was not controlled by the development team (Mohr 1987).
US version would eventually yield to programming techniques, and suggested that the group was merely adding to a tested and nearly completed software package.224

Despite these wishes, various project meetings throughout December 1995 were marked with increasing concern about the remaining changes and the inability to achieve many of these goals. With little time remaining until the mid-January implementation of EMPOWER in Community One, suggestions were made to disable features that were bugged or had data irrelevant to the Canadian setting 225 (Gibbs 1994; Keil 1995). In addition, plans were made to ensure that the earlier modules were ready, and later modules would be developed during January and delivered to the community once available.

The following summarize the processes in this stage:

---

224 The developer and programmers found it especially difficult to explain these constraints and alternative approaches to the content team who had difficulty understanding the technical issues behind the problems. It was also difficult to explain broken promises.

225 Disabling non-functioning features was a method of avoiding poor software code (Gibbs 1994). This was a last minute attempt to deal with project escalation (Keil 1995).
Table 5-4: Processes in the “Pandora’s Box Opens” Stage

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploring the Code Difficult</td>
<td>EP</td>
<td>Program</td>
<td>Sep. 1995 to Present</td>
<td>Frustration and time consuming</td>
<td>Programmer, Developer</td>
</tr>
<tr>
<td>Major content problems</td>
<td>EP</td>
<td>Software functionality and interface</td>
<td>Sep. 1995 to Dec. 1995</td>
<td>Surprise and Struggle to shape</td>
<td>Content team, Developer, Programmer</td>
</tr>
<tr>
<td>Outsourcing including database connection fails</td>
<td>EP</td>
<td>Data management</td>
<td>Sep. 1995 - Present</td>
<td>Time consumption, frustration, failure</td>
<td>Developer, programmer</td>
</tr>
<tr>
<td>Project scope expanded</td>
<td>EP/ST</td>
<td>Software functionality</td>
<td>Aug. 1995 - Present</td>
<td>Changes piling up</td>
<td>Content team, Developer, Programmer</td>
</tr>
<tr>
<td>Team response and impact</td>
<td>EP</td>
<td>Software functionality</td>
<td>Sep. 1995 - Present</td>
<td>Stick with the software</td>
<td>UBC project team</td>
</tr>
</tbody>
</table>

5.5. **Squeezing the Most from Toolbook and EMPOWER**

Despite the complexity and software constraints, a number of changes were made to the US version of EMPOWER to improve its performance, functionality, and relevance. Development team cohesion, resources, and discussions amongst individuals produced a cohesive vision about core software objectives and functionality, and a stability in other
dimensions required to reshape the software.

A number of content changes increased the relevance of the software to mammography screening, user planning, and the Canadian and international setting (Earl 1989; Cooper & Zmud 1990; Orlikowski & Robey 1991). These changes included spelling, case example descriptions, references, and language to more closely reflect Canadian mammography screening goals and terminology. Recent bibliographic references to applications of the planning-evaluation model were included. In addition, relevant mammography screening articles published since the production of the US version had been collected, but time and data input constraints prevented extensive updating within the software.

Second, navigation in the US version was linear. To allow more random and “back and forth” access, individual module map pages were constructed, showing the structure of the module pages, and allowing direct access to these pages (Henderson & Kyng 1991; Landauer 1995). Changes in module one (situational analysis) to simplify data entry into a single “spreadsheet” screen have also dramatically reduced navigational complexity. A bookmark icon was also added to the toolbar as another aid to navigation and exploration without losing one’s previous page.

---

226 Changes to the bibliography achieved strategic purposes (Earl 1989). Changes to the bibliography and other content information also increased task-technology fit (Cooper & Zmud 1990). Institutional and “must have” changes included in the software for breast cancer and Canadian relevance (Orlikowski & Robey 1991).

227 Direct access to pages provided direct user access, similar to Henderson & Kyng’s (1991) discussion about tailorable and user designed systems. Direct access and navigation for easier and more productive work after initial entry, is stated in Landauer’s (1995) user-centered design.
Third, a few of the outsourcing strategies increased the relevance and power of the program. For example, the programmer produced a linkage between a graphing code library and Toolbook, thus allowing graphing of project data.\textsuperscript{228}(Attewell 1992; Orlikowski 1992b; Landauer 1995).\textsuperscript{229} In addition, summary reports can be loaded directly into a Windows-based word processor, MS Write, with minimal navigation by the user.

Finally, most software and content bugs were removed, providing minimal authenticity\textsuperscript{230}(Van De Ven 1986; Earl 1989; Orlikowski 1992b). Summary reports are printing most of the information recorded by the user, and additional suggestions are being provided in the reports.

The following table summarizes the processes involved in this stage:

\begin{table}
\centering
\begin{tabular}{|l|l|}
\hline
\textbf{Process} & \textbf{Description} \\
\hline
Learning and supply side barriers were partially overcome & (Attewell 1992). An attempt to build tools that are normally used in epidemiological analysis, matched some premises of the technology with the decision environment (Orlikowski 1992b). Better design, and a focus on users' tasks and productivity increased task-technology fit (Landauer 1995).\textsuperscript{228} \textsuperscript{229} \\
\hline
However, limits with data storage prevented us from extending the graphing into epidemiological data because data storage and updating was too slow. & \textsuperscript{229} \\
\hline
Minimal authenticity of the technology in use was achieved & (Van De Ven 1986). Some strategic aspects and promises were reached (Earl 1989). Premises of the technology matched funding agency and development team expectations (Orlikowski 1992b).\textsuperscript{230} \\
\hline
\end{tabular}
\end{table}
Table 5-5: Processes in the “Squeezing the Most” Stage

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>Stakeholders from Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content changes</td>
<td>OI/ST</td>
<td>Interface and resources</td>
<td>July to Dec. 1995</td>
<td>Research and typing</td>
<td>Content team</td>
</tr>
</tbody>
</table>

5.6. Visions of a New Software Constrained by Resources

As a result of constraints within the US version, thoughts about developing EMPOWER from “scratch” and moving to a new software platform were discussed throughout the project. However, as implementation and usage of the software drew nearer, the pressure to continue modifying the old version increased.

Late in December and early in January, the developer raised a number of concerns and criticisms of the US and modified versions of EMPOWER. He predicted that the software would be unusable as a planning tool in the communities (Cooper & Zmud 1990; Orlikowski 1992b).

This prompted a more systematic redesign and rethinking of the EMPOWER software. In the design documents, the developer once again raised the idea of a “means-ends” chain, consistent with the planning-evaluation model itself, to allow full access to information, and to

---

231 Despite changes to Empower, the technology was perceived as inappropriate for the task (Cooper & Zmud 1990). The education premise of the software was interfering with planning tasks (Orlikowski 1992b).
simplify data collection and programming\textsuperscript{232} (Earl 1989; Hammer & Champy 1993). This would allow adaptation and flexibility in the tool’s usage\textsuperscript{233} (Boland 1987; Henderson & Kyng 1991).

However, despite continued enthusiasm and some resources, the expectations of communities, users and funding agencies forced the team to continue improving the current version, at least to the point of making it workable for the field tests and for educational or training purposes\textsuperscript{234} (Kling & Scacchi 1982; Earl 1989; Orlikowski & Robey 1991).

The following table summarizes the processes in this stage:

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Technological Level and/or Characteristics</th>
<th>Time</th>
<th>Type of Interaction</th>
<th>People Involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer points to major problems</td>
<td>EP/ST</td>
<td>Functionality</td>
<td>Dec. 1995 to Jan. 1996</td>
<td>Surprise and Difficulty</td>
<td>Developer, project team</td>
</tr>
<tr>
<td>Alternative design visions</td>
<td>OI/EP</td>
<td>Data design, functionality</td>
<td>Jan. to Mar. 1996</td>
<td>Enthusiasm and idea generation</td>
<td>Developer, project team</td>
</tr>
<tr>
<td>Constraints keep us focused on old EMPOWER</td>
<td>ST/EP</td>
<td>Use of old EMPOWER</td>
<td>Jan. to Apr. 1996</td>
<td>Pressure and need to satisfy groups</td>
<td>Project team, funding agencies, users, communities</td>
</tr>
</tbody>
</table>

\textsuperscript{232} A generic package was seen as strategic with current and new goals for assisting health promotion planners (Earl 1989). A rethinking of the concept of planning was beginning (Hammer & Champy 1993).

\textsuperscript{233} In-formation in information systems design describes this process (Boland 1987). The generic version would be tailorable for many health areas through developer data, user data and means-ends design (Henderson & Kyng 1991).

\textsuperscript{234} Expectations from funding and community settings constrained work on the generic version (Kling & Scacchi 1982). Outside pressures and opportunities kept the team focused on fixing the old version (Earl 1989). The software needed to be viable for the planning setting given time, software and resource constraints (Orlikowski & Robey 1991).
5.7. **Case Conclusion**

An analysis of the processes, process stages and the overall ISD process will provide some insights into the connection between the four process theories.

5.7.1. **Process Analysis**

In terms of *mutual exclusivity*, the following table shows the process types in each process stage of EMPOWER’s development:

<table>
<thead>
<tr>
<th>Process Stage</th>
<th>Number of Processes</th>
<th>Process Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISD Overall</td>
<td>25</td>
<td>TI 1, 1 Shared, OI 7, 7 Shared, EP 21, 11 Shared, ST 12, 12 Shared</td>
</tr>
</tbody>
</table>

The following table shows the ISD research areas were attached to the process stages (see table 2-6):
Table 5-8: ISD Research Categories within Process Stages

<table>
<thead>
<tr>
<th>Process Stage</th>
<th>Process Types</th>
<th>ISD Research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Emergence</td>
<td>OI (3)</td>
<td>Strategic information systems planning</td>
</tr>
<tr>
<td></td>
<td>EP (5)</td>
<td>Evolution of technology; Alternative views; Predicament of design</td>
</tr>
<tr>
<td></td>
<td>ST (2)</td>
<td>Innovation; Adaptive structuration; Structural power</td>
</tr>
<tr>
<td>Getting Ready</td>
<td>TI (1)</td>
<td>System</td>
</tr>
<tr>
<td></td>
<td>OI (2)</td>
<td>Strategic information systems planning; Task technology fit</td>
</tr>
<tr>
<td></td>
<td>EP (3)</td>
<td>Alternative views; Predicament of design; Evolution of technology</td>
</tr>
<tr>
<td></td>
<td>ST (4)</td>
<td>Adaptive structuration</td>
</tr>
<tr>
<td>Pandora’s Box Opens</td>
<td>EP (7)</td>
<td>Technology divergence (Uni); Alternative views; Predicament of</td>
</tr>
<tr>
<td></td>
<td></td>
<td>design; Evolution of technology; Innovation</td>
</tr>
<tr>
<td></td>
<td>ST (1)</td>
<td>Adaptive structuration</td>
</tr>
<tr>
<td>Squeezing the Most from</td>
<td>OI (1)</td>
<td>Strategic information systems planning; Task technology fit</td>
</tr>
<tr>
<td>EMPOWER</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EP (3)</td>
<td>User involvement; Innovation</td>
</tr>
<tr>
<td></td>
<td>ST (3)</td>
<td>Adaptive structuration</td>
</tr>
<tr>
<td>Visions of a New EMPOWER</td>
<td>OI (1)</td>
<td>Strategic information systems planning; Task technology fit</td>
</tr>
<tr>
<td>Constrained by Resources</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EP (3)</td>
<td>Technology divergence (Uni); User involvement</td>
</tr>
<tr>
<td></td>
<td>ST (2)</td>
<td>Adaptive Structuration</td>
</tr>
</tbody>
</table>

Keeping in mind the subjective selection of critical events and processes, the following observations can be stated. First, the emergent perspective is linked to twenty one of twenty five processes in the development of EMPOWER. However, unlike Case One, the appearance of this emergence was not from external groups or internal politics, but from the complexity of redeveloping the US version, the many ideas and visions of the software, and the development team's expertise and interactions. Second, social technology and organizational imperatives were used to describe twelve and seven of the twenty five
processes respectively, but all of these processes were also described by another type; usually the emergent perspective. Social technology assisted in describing many of the unpleasant and unexpected constraints and learning of the US version. In addition, social technology provides insights into the team's continued revision of the US version. External pressure from groups expecting the software forced the continuation. Finally, the technological imperative participated in describing only one process.

Examining *special case* and *equal validity* explanations, the following table summarizes the shared processes:

**Table 5-9: Examination of Shared Processes**

<table>
<thead>
<tr>
<th>Process</th>
<th>Total Shared*</th>
<th>with TI</th>
<th>with OI</th>
<th>with EP</th>
<th>with ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>OI</td>
<td>7</td>
<td>1</td>
<td>-</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>EP</td>
<td>11</td>
<td>0</td>
<td>3</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>ST</td>
<td>12</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>-</td>
</tr>
</tbody>
</table>

* Note: that the "total shared" number is one less than the sum in the TI, OI, and ST rows because of a triple-coded process.

All of the processes described with an organizational imperatives were also described with either social technology or the emergent perspective, indicating that strategy is most likely formulated within a set of constraints and/or opportunities presented by the context. These constraints and opportunities included the many visions and ideas, and the inability to fully understand and change the US software.

The eight processes shared between social technology and the emergent perspective may indicate a "fine line" between the two theories. Emergence was either a developer response to complexity and constraints (social technology), or an emergent shock that was
incorporated into later stages of development.

Examining *association* between process types over time, the following table summarizes the “following” patterns:

**Table 5-10: Number of Times a Process Type was Followed by Another Type**

<table>
<thead>
<tr>
<th>Process</th>
<th>Total</th>
<th>by TI</th>
<th>by OI</th>
<th>by EP</th>
<th>by ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OI</td>
<td>12</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>EP</td>
<td>34</td>
<td>1</td>
<td>7</td>
<td>16</td>
<td>10</td>
</tr>
<tr>
<td>ST</td>
<td>17</td>
<td>0</td>
<td>2</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>

The next table shows the likelihood of a *series* of processes with one type being terminated by one or more other process types:

**Table 5-11: Number of Times a Series of One Type was Terminated by Another Type**

<table>
<thead>
<tr>
<th>Process</th>
<th>Total</th>
<th>by TI</th>
<th>by OI</th>
<th>by EP</th>
<th>by ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI</td>
<td>2</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OI</td>
<td>10</td>
<td>0</td>
<td>-</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>EP</td>
<td>9</td>
<td>1</td>
<td>4</td>
<td>-</td>
<td>4</td>
</tr>
<tr>
<td>ST</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>-</td>
</tr>
</tbody>
</table>

Similar to case one, a process described by an emergent perspective was followed by another process described by an emergent perspective in 16 of 34 “followings”. This repetitive pattern did not appear with the other three theories. This may suggest that once an emergent perspective appeared, additional emergent processes, involving shocks and learning, followed.

Examining the “termination” data, a series of emergent processes were terminated equally by organizational imperatives and social technology processes. Emergence and social technology processes equally terminated organizational imperatives. On the other hand, social technology processes were overwhelming terminated by emergent processes (six of seven
times). It appears that processes described with a social technology perspectives rarely led to new processes described with an organizational imperative, while organizational imperatives and emergent processes led frequently to processes described by social technology, emergent or organizational imperatives. The story-line supports these findings. Beginning with emergent opportunities and the combining of individual interests, the project settled into organizational imperatives and social technological responses to those imperatives. Shocks from the environment and uncertainty around the software, however, hit the project during the middle of development. The group responded with new imperatives and/or incorporated new constraints and opportunities into design.

5.7.2. Information System Development

The ISD process overall is best described as emergent. Although the goal and specific means of producing the Canadian and international version of EMPOWER were driven by top-down plans, the problems with the US version, exploration, learning, testing, and numerous development ideas separated initial plans from the final outcome. The result was an extended educational version of the software, instead of a complete planning tool. Visions of a new and improved EMPOWER, using “means-ends” analysis and a new software platform, were constrained by resources.
6. Comparison and Integration

6.1. Introduction

Information systems development (ISD) involves a series of interactions or “processes” between technology and context over time. Each ISD project appears unique, given the different contexts, starting technologies, histories, processes and final technologies. However, the theoretical categorizing of critical technology-context processes allows comparison.

Chapter 6 compares and contrasts the two cases to reveal cross-case conclusions about technology-context processes. Important in this discussion are the stages and dimensions of stakeholders involved in ISD, the four process theories and the ISD research categories, the technological levels involved in a process, and the reconciliation approaches. The next four sections examine each of the four process theories across the cases. The final section integrates the four theories into a theoretical model.

6.2. Technological Imperative

To recall the definition, the technological imperative (TI) process theory dictates that technology impacts organization and context in a unidirectional manner. Developers have

---

235 Stages include initiation, adoption, adaptation, acceptance, routinization, and infusion. Dimensions include technology, users, tasks, organization, developers and suppliers, and environment.

236 The process theories include the technological imperative, organizational imperative, emergent perspective, and social technology. The ISD research categories, for example, included technology and systems for the technological imperative. See table 2-6 for a complete listing.

237 Methods of reconciling these fours views include mutual exclusion, special case, equal validity, and association.
little role in shaping or diffusing this technology, and users “adopt” and use the technology in a predictable manner. The following table summarizes the findings from both cases:

Table 6-1: Technological Imperative Summary from Case One and Case Two

<table>
<thead>
<tr>
<th>Case</th>
<th>Data</th>
<th>Shared with:*</th>
<th>Followed by:</th>
<th>Terminated by:</th>
<th>Technology Level</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3 of 43</td>
<td>OI 2 EP 1 ST 0</td>
<td>TI 1 OI 0 EP 3 ST 1</td>
<td>OI 0 EP 2 ST 1</td>
<td>All levels</td>
<td>Developer, funding agencies, coordinator</td>
</tr>
<tr>
<td>2</td>
<td>1 of 25</td>
<td>OI 1 EP 0 ST 1</td>
<td>TI 0 OI 0 EP 1 ST 1</td>
<td>OI 0 EP 1 ST 1</td>
<td>Infrastructure</td>
<td>Developer, US team, US company, UBC project team</td>
</tr>
</tbody>
</table>

Case One TI Literature: Technology and Systems
Case Two TI Literature: Systems

TI - Technological Imperative; OI - Organizational Imperative; EP - Emergent Perspective; ST - Social Technology

* - Note: The “shared with” number may be greater than the “data” column because of triple coded in a processes.

Little evidence supports a *mutual exclusion* of the technological imperative in explaining the data. The technological imperative *participated* in describing only four of the sixty eight critical processes in both cases. Three appeared in the SoftHeart case, and one in the Empower case.

There are two explanations why TI explained was not used often to describe the critical processes in the two case studies. First, longitudinal and historical analysis showed that many apparent technological imperatives, including EasySys and the US version of

---

238 Assuming each process represents an equal proportion of the data.
EMPOWER, were really emergent shocks or social technological constraints on the developing systems. For example, in Case One, the coordinator's requirement to create an interface in SoftHeart resembling EasySys was an attempt to keep intruders away from the software and her work.

Second, technology may also be more influenced by context during ISD than later stages of diffusion. For example, in Case One, the continuous changes in functionality, interface design, and database design demonstrate technology's subservience to context in both clinics. Possibly as the software is implemented and used, a technological imperative will appear.\(^\text{239}\)

Examining equal validity and special case patterns, TI's "sharing" with other process theories show three of the five processes were shared with an organizational imperative (OI), indicating a link between rational planning and technology impact. This TI-OI link is common in the strategic planning and structured information systems design literature (Kling & Scacchi 1982; McFarlan 1984; Earl 1989; Davenport & Short 1990). The two may be associated as either a two-step process, or have equal validity in explaining the same process. Technological imperatives appeared to be linked with an organizational imperative when the benefits of using the technology are perceived by many individuals (Earl 1989).\(^\text{240}\) For example, in Case One, the use of computers was reinforced by the perceived successful use of electronic patient records in other hospitals.

\(^{239}\) However, initial data on early implementation in both case studies show that the software has an incomplete and sometimes unplanned effect on context. For example, the use of Empower in the community has produced a limited focus on quality of life and health at planning meetings, and SoftHeart's deployment and usage in the clinics has resulted in a number of surprises.

\(^{240}\) See Landauer (1995) for a challenge to this view, showing that increases in IT per person is correlated with flat or decreasing productivity.
The stakeholders involved in the initiation of TI processes were mainly developers and suppliers of technology, and not organizational users and “management”. Most of these developer-derived technological imperatives were based on assumptions which were criticized by other stakeholders. Technological imperatives also included “legacy systems” that affected development (i.e. EasySys and the US version of Empower). “Legacy systems” imposed constraints and opportunities on design because the system is imbedded in the organization, and proved to be useful for the new designs (Petroski 1994). In case one, EasySys’ data has formed the basis of SoftHeart’s data. Without this data, SoftHeart would have been an empty shell with little data to support initial adoption and use. In case two, the US version of Empower dramatically influenced programming and design. Limited time and choice were key processes leading to technological imperatives (Thomas 1994). This suggests that TI may be a special case of social technology when the technological constraints are strict, time is limited, and/or the developer chooses to learn and take advantage of the legacy system (Orlikowski & Robey 1992).

The levels of technology correlated with a technological imperative ranged from system ideas to interface details. In case one, initial ideas of the developers and initiators of technology were technological imperatives. The redevelopment of EasySys’ interface was also a technological imperative, and was initially agreeable to the developer because the coordinator appeared to know her work and the computing tools required to complete that

241 Users in both case studies were unaware and uninvolved during early conception.

242 The human-computer interaction and framing literature infrequently discuss the boundaries of their findings. Regardless, the developer and his thesis committee also extended the findings into more complex outcomes such as patient health.
work. Further discussions with the coordinator, however, resulted in “requirements” to mimic keystroke-for-keystroke every interface characteristic in EasySys, uncovered political activity to block SoftHeart’s influence on EasySys, and revealed the coordinator’s intentions to stop the project.

Examining inter-process associations, most processes described as technological imperatives were shortly followed by or the result of processes described as emergent. For example, early funding proposals, based on technological imperative visions of multimedia education and information presentation affecting patient outcomes, were rejected by unconvinced health groups. As another example of TI’s association with the emergent perspective, the developer’s threat to quit the project prompted the Clinic One to fire the coordinator. This released the project from the EasySys technological imperative, and launched an emergent and chaotic phase of development.

6.3. Organizational Imperative

The organizational imperative (OI) theory mandates that organizations use a rational planning process to choose and/or develop technologies that will meet their strategic objectives. Developers implement technology identified by top management or users, and users adopt the technology as planned. The following table summarizes the findings from both cases:
Table 6-2: Organizational Imperative Summary from Case One and Case Two

<table>
<thead>
<tr>
<th>Case</th>
<th>Data</th>
<th>Shared with:*</th>
<th>Followed by:</th>
<th>Terminated by:</th>
<th>Technology Level</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15 of 43</td>
<td>TI 2</td>
<td>TI 2</td>
<td>TI 1</td>
<td>All levels</td>
<td>All stakeholders</td>
</tr>
<tr>
<td></td>
<td>13 Shared</td>
<td>EP 7</td>
<td>OI 7</td>
<td>EP 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ST 5</td>
<td>EP 8</td>
<td>ST 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>7 of 25</td>
<td>TI 1</td>
<td>TI 0</td>
<td>TI 0</td>
<td>All levels</td>
<td>All stakeholders</td>
</tr>
<tr>
<td></td>
<td>7 Shared</td>
<td>EP 3</td>
<td>OI 1</td>
<td>EP 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ST 4</td>
<td>EP 7</td>
<td>ST 4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Case One OI Literature: Strategic information systems planning; User involvement; Task technology fit

Case Two OI Literature: Strategic information systems planning; Task technology fit

TI - Technological Imperative; OI - Organizational Imperative; EP - Emergent Perspective; ST - Social Technology

* - Note: The "shared with" number may be greater than the "data" column because of triple coded in a processes.

Little evidence supports a *mutual exclusion* of the organizational imperative in describing the processes. In total, the organizational imperative *participated* in explaining twenty two of the sixty eight critical processes in both cases, and exclusively described only two critical processes. Fifteen participated and two exclusively described processes in Case One, and seven participated and none exclusively described processes in Case Two.

The developer-led nature of both projects weakened an organizational imperative description of the data. The early disinterest and disregard in the SoftHeart project amongst clinic directors and personnel weakened an organizational imperative description of this stage. In fact, most of the ‘strategic’ considerations of the software by “management” have appeared near the end of the development phase, as the two clinics understood SoftHeart and its potential importance to future operations (Thomas 1994). In both cases, the systems were
“pushed” by developers rather than “pulled” by users and management (Attewell 1992; Petroski 1994). Now that the software has proved it can work, most of the clinic personnel are discussing the future of SoftHeart for increasing the profile of the clinic to hospital, government, and research groups (Petroski 1994).

Examining equal validity and special case patterns, OI’s “sharing” of process descriptions with other process types demonstrated that strategic and IT decisions could not be divorced from the day-to-day realities of organizational life and computing technology. Twenty-four-three percent of the processes were shared with the emergent perspective and social technology in Case One, and seven of seven in Case Two. There are a number of descriptive examples of OI’s connection with EP and ST. Case One’s user-centered design led to many visions of the organization, the work, and the role of IT. Organizational imperatives incorporated new constraints and ideas to ensure the software’s relevance to the stakeholders. In case two, the “make” vs. “buy” decision was best described as an organizational imperative constrained by money, time, and uncertainty.

Examining the level of technology and time, the data demonstrated that organizational imperatives described processes affecting all levels of technology, but their effects are short-lived as emergent learning and shocks shook the project. In addition, the stakeholders involved in strategic direction included only the developers during early stages, thus...

---

243 This contradicts the normative separation between design and implementation advocated by some design philosophies.

244 Other reasons for the weak organizational imperatives may be the complexity of the context in Case One, and the relative novelty of the software in Case Two. Also, the developer’s inexperience with the clinical setting, electronic patient records and networked software may have prevented the project from developing top-down approaches to development. However, by claiming that the experience of the developer affects strategic planning implies that the emergent perspective accurately explains the process.
weakening the use of OI descriptions of the processes that focus on top-down strategic planning.

The organizational imperative literature attached to technology-context processes included strategic information systems planning and task-technology fit in both cases, and user involvement in Case One. This reflected the different approaches to development in the cases. Case One had extensive user involvement, and Case Two had very little. Some of the user involvement in Case One was categorized as an organizational imperative because the involvement was an attempt to build an integrated understanding of the organizational goals and information system.

Examining inter-process associations also demonstrated how context influenced organizational imperative planning. Most organizational imperatives were shortly followed by processes that were categorized as emergent perspectives or social technology. In Case One, thirteen of twenty-two “followings” were by emergent or social technology processes, and eight of nine OI “terminations” were by emergent and/or social technology processes. In Case Two, eleven of twelve “followings” were emergent or social technology, and all ten “terminations” were by the emergent or social technology perspectives.

Both cases provide tentative conclusions about the OI and EP/ST link. In Case One, periods of calm and agreement between various groups resulted in a number of initiatives driven by rational planning and development. For example, a developer-led strategic plan created the original vision of SoftHeart: to improve the communication between patient and provider to improve patient health. The specific “means” for achieving that end, however,
changed rapidly with user involvement, environmental shocks, the removal of the coordinator, and the developer’s changed understanding of the environment.\textsuperscript{245} In addition, user involvement from clinic personnel directed the project toward new “ends”, including quality of working life (Bostrom & Heinan 1977). The developer rationalized these administrative and research goals as contributing to patient health, although more indirectly than the clinical presentation of information during provider/patient discussion.\textsuperscript{246} The interaction between the core concept and IS means was bi-directional.

Case Two also illustrated that the original strategic plan was soon overridden by technological struggles with the US version of the software and external group pressure to produce a product. The project started with a general goal of producing a tool to assist the planning of breast cancer interventions. However, the project was driven by problems in the US version of the software, because the project team had assumed that most of the strategic planning and design were embodied within the software. Further exploration revealed these assumptions incorrect, but time and money were limited for redeveloping EMPOWER with a new software platform.

\textsuperscript{245} Even the use of computing technology was dropped in one proposal, in order to address the cost issues of computing equipment raised by the health groups.

\textsuperscript{246} Presently, computers have not been installed in the examination rooms to test the usage of computers during provider-patient discussions. Physicians and clinic personnel feel the technology will disrupt their discussion with the patient. This has occurred despite earlier agreements and discussions project establishing the clinicians’ interest in using computers in the examination rooms.
6.4. Emergent Perspective

The emergent perspective theory (Markus & Robey 1988) states that a complex interaction occurs between context and technology over time. In its stronger form, EP claims that technology and context are inseparable. Every context and technological appropriation and adaptation is unique. Developers' and users' ideas and visions determine the method of selection and criteria, the choices, and the implementation approaches (Thomas 1994). The following table summarizes the findings from both cases:

Table 6-3: Emergent Perspective Summary from Case One and Case Two

<table>
<thead>
<tr>
<th>Case</th>
<th>Data</th>
<th>Shared with:*</th>
<th>Followed by:</th>
<th>Terminated by:</th>
<th>Technology Level</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31 of 43</td>
<td>TI 1</td>
<td>TI 1</td>
<td>TI 1</td>
<td>All levels</td>
<td>Developer</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>OI 7</td>
<td>OI 11</td>
<td>OI 5</td>
<td></td>
<td>and new</td>
</tr>
<tr>
<td></td>
<td>Shared</td>
<td>ST 8</td>
<td>EP 23</td>
<td>ST 4</td>
<td></td>
<td>stakeholders</td>
</tr>
<tr>
<td>2</td>
<td>21 of 25</td>
<td>TI 0</td>
<td>TI 1</td>
<td>TI 1</td>
<td>All levels</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>OI 3</td>
<td>OI 7</td>
<td>OI 4</td>
<td></td>
<td>stakeholders</td>
</tr>
<tr>
<td></td>
<td>Shared</td>
<td>ST 8</td>
<td>EP 16</td>
<td>ST 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ST 10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Case One EP Literature: Technology divergence; Designer's views; Predicament of Design; Evolution of technology; User involvement; Innovation

Case Two EP Literature: Technology divergence; Alternative views; Evolution of technology; Innovation

TI - Technological Imperative; OI - Organizational Imperative; EP - Emergent Perspective; ST - Social Technology

* - Note: The "shared with" number may be greater than the "data" column because of triple coded in a processes.

Evidence does support EP's value in describing the data. The emergent perspective described fifty two of the sixty eight critical processes in both cases, and exclusively described
twenty six processes. In Case One, thirty one participated and sixteen exclusively described critical processes. In Case Two, twenty one participated and ten exclusively described critical processes.

Examining the "shared" processes revealed equal validity and special case patterns with other process theories. EP was closely allied with the organizational imperative and social technology perspective in describing processes. Eight of fifteen shared processes in Case One, and eight of eleven shared processes in Case Two were shared with social technology. Seven of fifteen in Case One, and three of eleven processes were shared with the organizational imperative. The linkage of the emergent perspective with social technology and organizational imperatives is also supported by inter-process examination of the "termination" data, with nine of ten Case One and eight of nine Case Two terminations resulting from organization imperatives or social technology.

These data suggests a descriptive fine line between the emergent perspective, the social technological perspective, and the organizational imperative. Emergence may be a "general case" of social technology where the boundaries around either organization or technology are weakly defined or changing. For example, once new stakeholders and knowledge of these stakeholders and there needs were identified and incorporated in the software's design, a period of social technology appeared before new shocks and discoveries sent the project into emergence.

Although the theoretical patterns are similar between the two cases, the emergent literature attached to the two cases differed. Case One included designer views, predicament
of design, and political and radical “bottom-up” user involvement. Case Two included alternative views and the evolution of technology. These differences are explained by different approaches to design, including the complex and fragmented setting of Case One, the exclusive control of development by the developer, and the extensive user involvement during design.

Reasons for the descriptive power of emergence with the critical technology-context processes are numerous. First, both projects emerged from an intersection of various individuals with different interests and expertise. Projects arose more from the wants and desires of developers than from organizational need. Experience in other fields of technological innovation suggests that supply-led development is fairly common, emergent, and based more on supplier desire and intuition than obvious societal need (Petroski 1994; Thomas 1994). In Case One, the developer’s database background and work experience provided a “subsidy” to the Case One’s initiation and creation, and Case Two’s development (King et al., 1994). The developer’s lack of experience with clinic settings required him to use various techniques and perspectives to develop an IS intervention (Klein & Lyytinen 1985; Boland 1987). Certainly, the developer’s learning, involvement, and interpretation of the setting and its potential effect on the project influenced development (Attewell 1992).

The onus was also on the developer to translate user comments into the technology. This process may be called an “inferential leap” (Turner 1987), because the exact reasoning and choice of interface and database design is difficult to describe, and based on compromise between competing objectives (Petroski 1994). The result of this “leap” is tested back in the environment through prototype demonstration, that revealed and uncovered new requirements
and opportunities, and ultimately sent development into another round of emergence (Schon 1983).

Second, environmental shocks, learning, discussion with new users, limited time and money, and prototype creation and demonstration based on user discussions, obscured the connection between strategy and technology. For example, Case One began in July 1992 as an experiment to test the effects of multimedia information on patient outcomes. It ended with the development of an electronic patient record that is currently implemented in clerical and research offices. Similarly, the EMPOWER project also experienced emergence. Many of the ideas and suggestions were not reflected in the software's design because the development team was unable to develop new functionality in the US version.\(^\text{247}\)

Third, both cases demonstrated that the 'ends' of the system are difficult to plan in advance. With over 15 users in Case One with various degrees of experience and professional background, SoftHeart was taken in many directions. The EMPOWER project faced emergence of a different kind, where the ends of the project were changing rapidly based on examination of the program, and as the 'wish list' continued to grow. In addition, findings from the field were to inform revision of the software in accordance with the funding agreement.

Fourth, technology can unexpectedly influence unplanned tasks and environmental dimensions. Many of these effects cannot be determined in advance.\(^\text{248}\) These unexpected

\(^{247}\) Examples include the graphing of epidemiological data and the construction of an improved database engine using the Database connection to fix the contact manager and other data intensive functions.

\(^{248}\) The literature on user centered design, development, and deployment states that the best test of a software product is a test with potential users. Logic, deduction, and intuition are not enough.
influences can produce new opportunities as well as new constraints that must be understood and worked into software design. The best approach to development in this situation is prototyping. For example, case one required detailed prototyping and understanding and testing of SoftHeart to determine how this “intrusion” would affect data entry requirements in the front office, physician time in searching for patient names during a consultation, report delivery if reports were printed on a network printer, information “dependency” during clinic visits if the server crashed, and provider discussion with a patient using a computer. The lack of early prototyping and testing in Case Two may have been partially responsible for some of its usage problems.

Finally, unexpected technological constraints in both cases affected interface, functionality, and strategic direction. In Case One, the struggle learning the Foxpro language, and the requirement to develop an interface similar to EasySys, produced emergence. In case two, the struggle with the US version prevented us from thoroughly addressing the original goal of fully supporting the planning of mammography campaigns.

6.5. Social Technology

The social technology (ST) theory suggests a “soft-line determinism” between technology and organizational structure. Equal focus is provided to both the constraining and enabling features of a technology, and the organization. Developers build technology within

---

249 Only extensive experience with the setting and the technology can allow IS planning using a top-down approach. This experience depends not so much on the age of the technology, but the experience of users, developers, and suppliers involved in the project. But even then, each setting is complex, requiring testing and feedback to build a local “theory” of the setting. Thus, experience can only serve as an analogue for future development projects.

250 What has been developed might be viewed as a prototype for field-testing.
constraints and opportunities. Users "adapt" the technology within technological, environmental, and user constraints.

Table 6-4: Social Technology Summary from Case One and Case Two

<table>
<thead>
<tr>
<th>Case</th>
<th>Data</th>
<th>Shared with:*</th>
<th>Followed by:</th>
<th>Terminated by:</th>
<th>Technology Level</th>
<th>Stakeholders</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16 of 43</td>
<td>TI 0</td>
<td>TI 1</td>
<td>TI 1</td>
<td>All levels</td>
<td>Developer and new stakeholders</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>OI 5</td>
<td>OI 5</td>
<td>OI 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ST 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12 of 25</td>
<td>TI 1</td>
<td>TI 0</td>
<td>TI 0</td>
<td>All levels</td>
<td>All stakeholders</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>OI 4</td>
<td>OI 2</td>
<td>OI 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ST 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Case One ST Literature: Adaptive structuration; Structural power
Case Two ST Literature: Adaptive structuration

TI - Technological Imperative; OI - Organizational Imperative; EP - Emergent Perspective; ST - Social Technology
* - Note: The "shared with" number may be greater than the "data" column because of triple coded in a processes.

Little evidence supports a mutual exclusive ability of social technology in describing critical processes. In total, the social technology perspective participated in describing twenty eight of the sixty eight processes, and exclusively described four processes. In Case One, sixteen participated and four exclusively explained processes. In Case Two, twelve participated and none exclusively explained processes.

Examining the shared processes, ST is closely allied with the organizational imperative and emergent perspectives. The "followings" and "terminations" also illustrated a connection between social technology and the emergent perspective. In terms of time and association, the social technology perspective, in combination with the emergent perspective, described
certain processes such as idea and software adaptation to external demands and shocks. Social technology also helped in understanding the learning of the software required in both case studies, and the opportunities of expanding EasySys in Case One. Thus, both cases exhibited social technology perspectives after emergent processes had settled down, and the development team was able to incorporate these constraints and opportunities into the project. This implies a possible close relationship between social technology and the emergent perspective.

There are a number of reasons why the social technology perspective "shared" in describing many critical processes or was associated with the organizational imperative and emergent perspective. First, technological and resource constraints and opportunities appeared to affect the vision and implementation of the information system (Argyris, Putnam, & Smith 1985). It therefore affected any strategic planning and organizational imperatives. For example, faced with little funding and resource support to support the SoftHeart project in May of 1995, the developer decided to use some of his own funding and programming skills to produce an extension to EasySys in order to build graphical and risk reports for patients and physicians.²⁵¹

Second, Case One also demonstrated that people are the builders and enforcers of constraints and opportunities, and that the loss of these individuals produces emergence or imperatives. For example, in Case One, many of the constraints by the first coordinator

²⁵¹ That decision has led me to at least one year of design and programming involvement (much of it full time) in SoftHeart's development.
appeared to be solid and permanent. However, her departure released many of these constraints, producing a period of emergence.

Finally, as mentioned in the technological imperative section, both cases were strongly influenced by legacy systems. EasySys provided many constraints for design in early stages. Initially, this constraint was accepted by the developer because the program captured valuable data that could be reported and presented effectively. In addition, with the developer's limited clinical experience, the program allowed him to learn a great deal about the types of data and the form of that data important to the clinic. Despite the firing of the coordinator, EasySys influenced the database structure and look-and-feel of SoftHeart.

The US version of EMPOWER, a legacy system, and the environment around the development of EMPOWER also enabled and constrained development. The development team's choice to enhance the US version, in order to build upon what the development team considered to be a stable product, introduced a set of known and unknown constraints on development. Initial results of software testing in the communities indicated that the software did not meet the planning needs of the communities, prompting discussions of building EMPOWER "from scratch" using a different software platform. However, the lack of time and money, the push to produce a Canadianized product to satisfy cooperating agencies, users anticipating the new software, and two waiting communities wishing to test the software, prevented the move to a new software platform.

Most technological levels were subjected to social technology, especially the functionality and infrastructure. Database and multimedia applications are supported by a large supplier and expertise base that is well established in the microcomputer industry. In
case one, the program and data structure were constrained by industry conventions and ideas. In addition, the interface characteristics and input approaches in both cases were constrained and enabled by the Windows environment.

*Stakeholders* most involved in social technology processes were the developers, suppliers of software and IT, the thesis committee, and funding agencies. Each provided certain enabling and/or constraining influences on the options available. First, the developer’s background and experience both enabled and constrained both projects. Second, suppliers of software enabled and constrained development, and explained the constraints of legacy systems software. Third, the developer’s committee provided access to Clinic One, alternative ideas, avenues of development, and knowledge of the IS and health fields. Finally, funding agency ideas and responses also altered the direction of Case One and Two.

In addition to the *adaptive structuration* literature found in both cases, Case One included some *structural power* literature to help understand and describe critical processes. The clinical setting produced somewhat predictable patterns in personnel responses to IT ideas. For example, the reshaping of SoftHeart toward research and administration and away from the original goals of direct patient care, is the result of structural and funding characteristics in both clinics.

Important in most social technology processes was uncertain and imperfect *choice*. Choices constrained and enabled technological development in Case One (Argyris, Putnam, &

---

252 The "medical record" and its data structure and data elements have reached certain standards.

253 These conventions are themselves the result of emergent processes in the computing industry. For example, the Windows interface is based on an emergent history of research done at Xerox Parc that was eventually incorporated into the Macintosh operating system, which then spread through Microsoft’s growing dominance into IBM PCs.
Choosing Clinic One as a site enabled certain types of IS development but constrained others. Choosing Windows Foxpro versus DOS Foxpro allowed the developer to tap the many windows resources and look-and-feel, but reduced the processing speed and introduced some common Windows software bugs into the program.\(^{254}\) Many of these choices were made without realizing the full set of constraints or enablers on future development.\(^{255}\) In case two, many of the problems with the US version were not fully discovered until later in the project.

If some choice constrained, other choices weakened and reduced a constraint. For example, to deal with the slow graphing problem in Case One, the developer purchased a graphics package that linked directly into Foxpro as a library of commands. As another example, to address the criticisms of computer cost in the first proposals, the developer changed future proposals to include only a single computer to print enhanced reports for providers and patients.

Nevertheless, constraints were often beneficial if they guided and protected the project from over-expansion and streamlined development. Constraints were also good if the software had already been constructed considering these constraints, but a problem if new constraints forced redevelopment. For example, too little constraint in the SoftHeart project led the project in many directions after the coordinator was fired.

\(^{254}\) The most famous is the General Fault Protection error (GFP) resulting from Windows poor memory management capabilities.

\(^{255}\) I wonder if I would have known the problems and constraints I would face during SoftHeart's development before beginning, whether I would have even started.
In conclusion, social technology appears to be associated and possibly a special case of emergence where the boundaries around either organization or technology are more strongly defined and stable. These constraints are incorporated into design and/or changed by new choices or environmental shocks.

6.6. Integrating the Four Theories

The previous discussion has revealed some tentative associations and patterns between the four process theories. This section introduces a first attempt at an integrated model of context and technology interaction. Two contextual qualities appear to "correlate" with the four process theories, given the level of technology, task(s), and stakeholders involved. The first quality is the agreement about the use of technology for a task amongst those stakeholders who are essential to the technology's operation. Agreement on the use of a specific technology level or function can be classified as either integrated and clear or diverse and vague. The second contextual quality is the potential adaptation of the technology for use and development. Technological change may best be described as either nonadaptable or adaptable.

In addition, the model attempts to include dynamics and change by showing technology being composed of several layers of technological levels attached to various tasks, and time as mixing and changing the stakeholders, tasks and technological levels considered. Time and technological levels provide a tentative but strong reason why emergence best explained the critical processes in both cases.

The following figure summarizes the integration amongst the four perspectives:
As the figure suggests, *technological imperatives* appear when the technology or technological part for complex and multifaceted technology, is *nonadaptable* in terms of use and construction, and the stakeholders are *integrated* and *clear* about the use of that technology for a particular task. The *organizational imperative* implies context can *adapt* the technology through purchase or construction, and that stakeholders have *clear and integrated* ideas about the technology’s use for a task. The *social technology* perspective straddles the two dimensions, suggesting that technology is *somewhat adaptable*, but boundaries limit this adaptation. In addition, agreement on the usage of the technology for the task is *somewhat integrated and clear*, but not entirely.
Finally, the *emergent perspective* appears whenever there are *diverse and vague* agreements about the use of the technology for a task. Diverse agreement and nonadaptable technology yields emergence; in some cases rejection, avoidance, or "work arounds". In addition, diverse and vague agreement between stakeholders and adaptable technology, is also emergent. In this case, these views may be potentially incorporated into the different parts or levels of the technology. In both cases, the result is that the technology’s development and use is a complex interaction between the many stakeholders and the technology.

Note that the conclusions about a process depend on the *stakeholders*, the *part or level of technology*, and the *task* being examined. For example, an organizational imperative may explain the use of database applications, but the interface characteristics may be explained by emergence. The reason is that although the use of databases may be integrated and clear, the design of the interface may reveal diverse and vague opinions. As an example of stakeholder agreement, a system that involves a single user will have more integrated and clear agreement than a system that involves many individuals from separate departments. The wider the technology and/or task, the more stakeholders will be involved in a process, and the greater the chances for disagreement.

*Processes* create and integrate various old and new stakeholders with certain technological levels and tasks in a constant reassessment of their use and fit. Change is introduced over time through *learning* and *environmental shocks* when new stakeholders, new technology, new tasks, new ideas, new agreements or conflict, or all five are added to a

---

256 The term "work arounds" describes the manual methods used around an information technology. In some cases, these work arounds may consume more time than if the entire process was done manually.
process. These processes will resemble one of the four process theories. *Impact* occurs when this mixing of old and new tasks, technologies, and stakeholders produces change. For example, EasySys represented a strong social technology and technological imperative before the developer of SoftHeart arrived. Despite disagreement about EasySys' use in clinic one, no one knew how to change the system, except for the coordinator and her programmer. The developer's involvement opened EasySys to modification, thus prompting conflict, and the eventual removal of the coordinator from the clinic. The result was an explosion in requirements as the diverse views expressed themselves into the requirements for the new software.

From the case data and the integrated theoretical model, a major conclusion can be stated. Because most system development occurs in relatively complex settings using advanced IS and IT, requires significant development time, and will involve many diverse stakeholders, most system development projects will be marked by at least one emergent process. Complex settings imply many stakeholders with different goals and means for achieving those goals, and considerable uncertainty in developing technology compatible with these views. Both complex setting and technology implies that *learning* of the context will occur throughout development, often making previous plans and development work irrelevant. In addition, longer development times implies that new technology, individuals, or tasks will appear, making current technology and plans obsolete. Emergent *shocks*, such as the closing of Hospital One in Case One, are a higher likelihood in longer system development.

\[^{257}\text{Only through efforts to achieve integrated views or to close the technology, did a more organizational and social technology perspectives appear.}\]
projects, also making plans and development work irrelevant. As a result, a major shift in the project design to accommodate emergent shocks will result in emergence overall.\textsuperscript{258} This claim is supported by comparing the initial plans to the final software product in both cases. Both cases and other cases of technological development support this hypothesis.\textsuperscript{259}

6.7. Conclusions

Processes in both case studies reveal a pattern among the four process theories. Chapter 6 has integrated the literature of chapter 2 with the data from chapter 4 and 5 to provide a theoretical discussion and integration. Data supporting each of the four process theories were discussed. \textit{Mutual exclusivity, equal validity, special case,} and \textit{associations} between the four processes were also discussed. \textit{Level of technology, dimensions,} and \textit{time} were explored to position and characterize the theories within the cases.

The result was an integrated framework placing the four theories within two contextual "qualities": \textit{stakeholder agreement} on the use of a technology for certain tasks (integrated/clear or diverse/vague) and \textit{technological change} (nonadaptable or adaptable). Technological imperatives appear when stakeholder views are clear and integrated, and technological development or use is nonadaptable technology. Organizational imperatives arise when stakeholders have integrated and clear views, and technology is adaptable. Social technological processes appear when agreement is somewhat diverse and technology

\textsuperscript{258} Technology-context interaction between these shocks may be best described as social technology or organizational imperatives.

\textsuperscript{259} Another more extreme example is the development of the automobile and the aircraft. The inventors of the combustion engine probably were not planning for car travel and air travel on such a massive scale, and the fuel production, road construction, pollution and other consequences that have accompanied these innovations.
somewhat adaptable. Finally, emergent processes emerge from diverse and vague agreement amongst stakeholders. If the technology is nonadaptable, then the system risks rejection, avoidance, or "work arounds". If the technology is adaptable, then the process will involve a complex and non-deterministic interplay between technology and context. Individual processes bring together old and new stakeholders, technologies, and tasks. A process takes on characteristics of one of the four theories, and the result of the process is impact: either the status quo or change.

A major conclusion of the thesis is that complex settings, complex IS/IT, and increased development time will likely produce significant learning and/or contextual shocks during development, requiring a major modification of development plans. As a result, comparing initial plans to the final product, the overall information system development process will resemble the emergent perspective.
Chapter 7: Contributions, Implications and Future Research

7. Implications and Future Research

This study of technology-context processes during ISD suggests a number of contributions and implications for researchers and practitioners. This chapter describes these contributions and implications, the limitations of the study, and future research directions.

7.1. Contributions

This thesis offers the following contributions to IS research and practice. First, a longitudinal study of the process of information systems development within two natural and complex settings has been conducted (Pettigrew 1979; Kling & Scacchi 1982; Das 1983; Pettigrew 1985; Vitalari 1985; Markus & Robey 1988; Barley 1990). As a result, the two case studies are "relevatory" cases (Yin 1994), examining the interaction between context and technology continuously during information system development. The thesis explored ISD in these settings over a significant time period; 4 years in Case One and 1 year in Case Two. This provides a historical analysis of ISD, a rich description of the people involved, and an understanding of their chronological influence on design.

Second, four theories of technology-context interaction were extracted from the IS and technology literature to classify a substantial amount of ISD research (Kling 1980; Kling & Scacchi 1982; Markus 1983; Van De Ven 1986; Markus & Robey 1988; Hirschheim & Klein 1989; Burton 1992; DeSanctis & Poole 1994; Myers 1994b; Thomas 1994). Technological imperative, organizational imperative, social technology, and the emergent perspective provided comprehensive coverage of the possible interactions between people and
technology. These theories provide four insightful "lenses" for interpreting the ISD research, and a rich theoretical and empirical background for interpreting ISD cases.

Third, these four process theories are integrated into a single theoretical model through two contextual "qualities": stakeholder agreement about the technology's use toward a task, and the adaptability of using and developing the technology (see Figure 6.1). Change was explained by the linking of old and new stakeholders, technologies, tasks, ideas, and environmental shocks.

Fourth, as a contribution to action research, the thesis portrayed the developer's role and influence on context and technology during ISD. It also demonstrated the use of newer methods for action research during ISD, such as hermeneutics, phenomenology, and critical theory (Susman & Evered 1978; Boland 1985, Lyytinen & Klein 1985; Boland 1987, 1990; Orlikowski & Baroudi 1991; Myers 1994b). In doing this, the thesis also contributes to a growing field of research joining research and practice in a constructive dialogue.

Fifth, the research provided a method of identifying processes affecting ISD, and the linking of these "situated" processes to the IS research. Using "technology level", time, and stakeholders involved, processes were uncovered, examined and linked to the ISD literature. These process characteristics were then used to reach conclusions about the process stage and the overall ISD process. The result is a useful way of drawing theory to guide practice, and practice to guide theory.

Sixth, extensive reading of the epidemiology and health promotion literature has identified areas where IT can support interventions for improving health. This will assist in building IT and IS that address patient and population health.
Seventh, the thesis work has transmitted two new “packages” (Kling 1980) into two settings. SoftHeart has begun early implementation in the two clinics, and is reaching a stable state of development and use. EMPOWER is providing the development team with information about its capabilities in two Canadian communities, and is signaling redesign areas that are to be incorporated into the current or future versions.

Finally, both cases have produced a rich set of data for future research. Diary notes from both cases exceeded 800 pages. Documentation of design notes, database schemas, reports, and meeting minutes exceeded 500 pages. Typewritten summaries of the diaries and documentation summaries produced 189 single-spaced pages of data for the SoftHeart case, and 57 pages for the Empower project. In addition, there are 15 prototypes of SoftHeart, and weekly backups of EMPOWER during the course of development. These data are resources for future research, including evolutionary design (Petroski 1994), testing alternative theoretical frameworks, and conducting variance research (Sabherwal & Robey 1995).

7.2. Implications

Many of the implications for research and practice point to a tension between two opposing forces during ISD and implementation that need to be considered and balanced.

First, emergence is a crucial component of system design. Complex contexts of development in both cases involved many individuals, groups, technological suppliers, developers, and technological options influencing design and development. Along with each individual and/or group came a series of different languages, goals, and technological means to achieving those goals. As a result, key decisions were not technical but social, especially the
decisions about what the technology would do and contain (Argyris, Putnam & Smith 1985). Emergent learning and shocks caused overall emergence as the developer or development team adapted the software.

It is perhaps initially distressing to admit that emergence is an important part of software development projects. However, emergence and an awareness and readiness for emergence was a requirement of good design. It is important for designers to alter plans, and design and build software that addresses learning, shocks and events. An understanding of emergence is essential for managing user expectations. In addition, the appearance of project emergence is a sign of innovation and adaptability, as unexpected tasks, technology, and stakeholders are brought together during development.

As a result of emergence, there is an inherent tension between the seizing of new opportunities and continuing with detailed plans. The solution is a balance between adaptation and restriction of further design changes. If a new opportunity is seized, the developer must also pause and rethink the basic organization and design of the software, to ensure new requirements are consistently applied to reduce program complexity.

Second and related to the first implication, organizational and technological imperatives are difficult to achieve. IS's ability to be interpreted differently by different groups and the onus on the developers of technology to meet stakeholder interests weakens these two imperatives. More prevalent in the cases are social technological processes that

\[260\] In Case One, the developer's original technical perspective on design made emergent shocks and revised learning difficult to absorb into the project.
appear before and after emergent processes, providing a set of learned constraints and opportunities for development.

Third and related to the cause of emergence, both ISD projects emerged from suppliers of technology. Developer visions and personality, design approaches, diverse personal views, inferential leaps from user comments to design, learning, system redesign and rework, and contextual shocks significantly affected the course of development. Other research points to supply-led innovation appearing as an upstart and challenge to inertia and busy demands in user organizations. This implies that a significant portion of ISD is driven by developers, not upper management who direct developers to implement requirements.\textsuperscript{261} As a result, software development places the onus on the developers to prove the worth of and to take responsibility for IS development.

Fourth and related to the cause of emergence, is the involvement of stakeholders in producing and being influenced by these emergent processes. The outcome of involvement will depend on the goals of various stakeholders, the technology, the task, and the characteristics of involvement. Especially important is the history of various groups and legacy systems.\textsuperscript{262} Choosing a “bottom-up” organizational imperative approach by involving users to achieve a consensus around the goals and technological means, or an emergent perspective by involving users and recognizing political opposition and “accommodation”, depends on this context. As a result, it is unclear whether system development projects

\textsuperscript{261} This is the “hammer finding the nail” issue, where those trained in a certain field use their skills to craft new solutions to old and new problems.

\textsuperscript{262} For example, the coordinator in Case One, despite her early involvement in the redesign, later blocked and discouraged SoftHeart’s development. She might have felt the project was going to reintroduce the research group into controlling her work. As a result, user involvement led to increased conflict and near failure of the project.
should: 1) build technology to accommodate different views (Hirschheim & Klein 1989; Checkland & Scholes 1990); 2) market technology to convince diverse views that different or similar parts of the technology can serve their needs; 3) side with one powerful party at the expense of other groups (Hirschheim & Klein 1989); and/or 4) move people to agreement through negotiation and integration. Deciding amongst these various competing approaches will depend on learning, the development stage, the political strength and views of various groups, the understanding of stakeholder views, and the adaptability of the software for development and use (Dutton 1981; Markus 1981; Markus 1983; Mohr 1987). What is clear is that the political and power structures in an organization and its connection with user involvement, must be consciously “managed” and assessed during system development.

Fifth, emergence either resulted from or was influenced by the use of prototyping. The alternative to prototyping, formalized methodologies, were inadequate in Case One, as demonstrated by their limited assistance and use during the project. It was difficult to plan detailed requirements and design documents with limited understanding of the general and specific clinical environment, and little understanding of the project and the analysis methods by clinic personnel. In Case One, prototyping was used instead, to allow testing of the Foxpro software, to build upon the wealth of data already available in EasySys, and to generate enthusiasm for working prototypes. In Case Two, the US version of Empower was a prototype, which was modified into another prototype for testing in the field. Prototyping allowed the exploration of context, and provided a method of testing certain ideas back into the setting. In both cases, reaction from users provided instant corrective feedback, increasing learning, and establishing trust. In addition, imperfect choice allowed the developers to
address multiple constraints, realizing that the full impact of these choices would be forthcoming. Any unpleasant outcomes were hedged with escape clauses.

Sixth and as a result of prototyping, both cases dissolved the distinctions between design and implementation, development and maintenance, and users and developers. Implementation was considered throughout design, given the multitude of possible designs that can be drawn from a set of needs. The designs were both “mentally” and physically tested in the context to determine their viability. Learning of new requirements and needs, and even problem definition were further refined for the next phases. The result was a cycle of continuous learning and iterative design.

Prototyping also dissolved the distinction between development and maintenance, because maintenance is a smaller iterative cycle of development. In addition, shocks from the environment and learning forced a wider cycle of development in later stages to ensure the software’s continued relevance to the organization.

Prototyping and user involvement also dissolved the distinction between developers and users. In a sense, prototyping and user involvement makes users into developers, and developers into users. More generally, it is this common link between two individuals with different goals that produces at least an understanding of each others’ perspectives that was crucial to successful ISD.

Seventh, there is a tension between developing a significantly different IS for an organization or incremental changes to the current IS. This tension is similar to Zuboff’s

\[263\] In these cases, I was also a user of the system in terms of the “use” logs and research information collected from both sites.
(1988) automating versus informing, Attewell’s (1992) competence-enhancing and competence-destroying, Petroski’s (1994) evolutionary innovation, Hammer & Champy’s (1993) “don’t automate; obliterate”, and Mohr’s (1987) routine change and readaptation.\footnote{264} Automation is concrete, easier to address and competence enhancing, but may produce only a limited impact on work and effectiveness. On the other hand, “informating” and transformation through information technology has a potentially larger impact on work and effectiveness. The wider boundaries of the new IS, however, will attract and require more interest groups than an automation project, and integrate many groups unfamiliar each other (Kling 1987). As a result, a wider boundary can potentially overrun the resources available to consult these groups, risking system obsolescence, restricted involvement, and system survival (Van De Ven 1986).\footnote{265} Once again, the solution is to find a balance between ambitious IS development that causes widespread change, and minimal ISD that changes very little.

Regardless of the outcome, a balance is required between the “analog” (organization setting) and “digital” (computer) world. Determining the tasks to be supported by computerization and tasks that would be harmed by computerization is difficult and requires constant monitoring. The blind overuse of computers into every human activity is difficult at best and harmful at worst.

\footnote{264} It also suggests that the obliteration of organizational processes from those far removed from the task may be dangerous and based on naive assumptions about the value of an organizational task (Hammer & Champy 1993).\footnote{265} Argyris, Putnam, & Smith (1985) comment that single-loop learning (automation) is much easier than double-loop learning (informating).
7.3. Limitations and Future Research

Limitations of case research are based on the four validities: internal validity, external validity, construct validity, and reliability (Yin 1994). These limitations form the basis for future research.

Crucial to many of the conclusions are the use of the four process theories, the subjective categorization of ISD research into one of the four theories, the selection of critical events and processes, and the connection of technology-context processes to the research (construct validity and reliability). The integrated theoretical model linking the four context-technology theories together, also relies on subjective interpretation.

Future research with the case data, examining a more systematic coding and analyzing of the diaries, documentation, and design notes will increase construct validity. This data will also be examined using other development theories of Kling and Scacchi (1982), Argyris, Putnam, & Smith (1985), Hirschheim & Klein (1989), which will provide different theoretical lenses on the results. Data will also be analyzed using variance and statistical research to examine process variables and their correlation over time. This will also increase internal validity (Sabherwal & Robey 1995). Other data collected from the cases, including various prototypes of SoftHeart and EMPOWER, can be examined for confirming and extending the theoretical model.

266 Discrete entity and web models.
267 Single-loop and double loop learning: I, O-I, II, and O-II models.
268 Functionalist, social relativism, radical structuralism, and neo-humanism.
Chapter 7: Contributions, Implications and Future Research

Future cases are also required to fully test the integrated model to overcome the exploratory nature of this thesis. This will increase the internal validity and improve on the construct validity and measurement of concepts such as dimension, process, process stages, technological imperative, organizational imperative, social technology, emergent perspective, technology, power, and organizational structure (Yin 1994). The use and/or development of a system methodology closely related to the theory may also be examined (Checkland & Scholes 1990), to test and refine of the model and the methodology.

The other weaknesses of the thesis relate to the external validity of the cases. The results of two complex case studies must be carefully generalized and researched in other settings. These future settings should include differences in all six contextual dimensions including organization, user groups, technology, environments, and developer/suppliers.

In particular, further observation of the later implementation stages in both case studies will provide a continued examination of the later implementation stages and ‘impact’. In Case One, this includes patient/provider talk, and changes in patient attitudes, intentions, social support, reinforcement, patient behavior, biochemical outcomes, and possibly disease incidence onset. In case two, ‘impact’ will include the health promotion planning process, the health promotion plan and implementation, changes in mammography rates, and diffusion of EMPOWER into other settings. The data from later stages of implementation may demonstrate more technological and organization imperatives.

---

269 Although data were collected from a second case to help confirm the initial framework, the revised theory is the product of both case study analyses.

270 Construct validity and internal validity have been increased in the thesis by using multiple sources of data, and time series analysis. Reliability has been increased by providing the case descriptions, and a detailed methodology provided in Chapter 3.
Weakening the generalizability of the results is the developer's background, experience, and role in other projects. My development role in case one and two gave me a special lens on the results of the development. Although individuals in the case studies confirmed my findings (Yin 1994), these other individuals may have seen and experienced other events that I was unable to experience. I also had an important interest and effect on the course of the case studies and my interpretation the information must be taken into consideration. Future research will need to use observational and survey approaches to assess independently the theoretical model.

Future research beyond the two case studies can include comparison of user, management, and developer impressions during development. Different types and conditions of user involvement may be examined to study more closely the process of user involvement in action research. Finally, the social conditions of technological imperatives and organizational imperatives could be examined.

7.4. Concluding Comments

This thesis joins a stream of research longitudinally studying the process ISD, and the interaction between technology and context. The two cases will continue to be studied into the later stages of implementation, with the assistance of postdoctoral funding.

Information system development involves a series of complex processes between technology and context. As a result, the future of both context and technology is fortunately non-deterministic. This allows for numerous ideas and visions to be incorporated into and influenced by technology. I hope to observe, facilitate and contribute to this interaction.
8. References


References


References


References


References


References


References


References


References


References


9. Appendix A: Events in the Development of SoftHeart

SoftHeart was developed over a period of time and progressed through a series of events, processes and stages of development. This appendix reports on the events among individuals, myself and the software during the development of SoftHeart. These events will be discussed within the following stages: project emergence, health funders skeptical, our response, old and new environmental shocks, decision to move ahead, prototype demonstration sparks interest and constraint, conflict erupts, decision to act, project builds momentum, development quagmire, and recovery and slow growth to early implementation. Note that these stages, although presented sequentially, were overlapping to some degree. The final section discusses what lies ahead.

9.1. Project Emergence

The project emerged from a mixture of individual interests, skills, and experience. My background influenced my attraction to the project and its original focus. My interest in integrating information systems and health care started with my undergraduate education. During that time, I became interested in accounting, health economics and organizational theory, and produced an undergraduate thesis examining case mix groupings in hospitals; a standard costing method of categorizing patients by disease (April 1990). My database experience provided me some of the skills for developing
SoftHeart, and also certain perspectives on system development. The training of end-users in a company established by my partner and myself gave me a glimpse of computing "in the trenches", and its effect on the companies and end-user work (May 1990 - February 1991).\textsuperscript{273} In addition, my experience as a computer programmer hired by the Finance department at a fibre optic cable plant, led to some insights and perspectives about computing and work. I was secretly hired by the Finance department to deal with its programming needs due to the MIS department's inability to meet these needs.\textsuperscript{274} The challenge of developing reports under conditions of limited documentation, time pressure, and MIS's watchful eye led to certain beliefs about end-user computing in large organizations.\textsuperscript{275} Finally, a number of incidents revealed the negative effects of computing on working life.\textsuperscript{276}

The SoftHeart project was an opportunity for myself, and three researchers, including one clinic physician, to achieve numerous goals.\textsuperscript{277} The project emerged out of my interest in health and information, a need to find an interesting thesis topic, and a

\begin{itemize}
\item \textsuperscript{273} I talked to many users who developed some interesting and creative applications of generic software, illustrating how software is creatively adapted for individual needs. One user used Lotus 1-2-3 for word processing. The documents looked pretty impressive.
\item \textsuperscript{274} Human resources had just received the hiring papers the day I arrived.
\item \textsuperscript{275} MIS's centralized approach to computing violated many of my beliefs about end-user computing.
\item \textsuperscript{276} At that time, the fibre optic plant was an unwanted division in the corporate company, and had received numerous cuts in funding. The firing of a billing person one week after developing a program that reduced her time by 8 to 10 hours per week, pointed to the dark side of computing spurred by cost cutting measures. Employees at the firm mentioned that this clerk had received poor evaluations for some time, and that the program had nothing to do with her firing.
\item \textsuperscript{277} The name "SoftHeart" was not attached to the project until July 2, 1995. In fact, the project did not have a name until June 1994 when the software was called the "Adherence Application" to reflect its focus on increasing patient adherence to treatment regimens.
\end{itemize}
chance for three colleagues to carry out research together (my supervisor, one committee member, and the clinic physician).278

During late 1991 and early 1992, I explored various research questions and ideas such as IT’s poor diffusion amongst physicians, the poor match between user needs and IT, effects of IT on organizational structure, and the use of information to improve genetic screening and economic outcomes of information. My first meeting with my future supervisor came on February 1, 1992. I explained my interest in health care and its effect on organizational structure. At that time, I was interested in using an ethnographic approach to study the historical process of IT development and adoption in hospitals.279 My supervisor and his colleagues were interested in the economic effect of information used to assist screening of patients, based on their discussion with the Clinic One physician.

Trying to consider how to combine my interest and theirs, I examined the framing literature that I had learned in a course the previous semester, and thought of trying to use that literature to develop information that would enhance patient understanding and motivation toward the information (March 1992 - May 1992). This concept of information framing was further extended with literature on multimedia computing and graphical presentation. My visit to Clinic One in July of 1992 seemed to suggest that information framing and multimedia computers could be used by patients while waiting to see

278 All three had met a number of months before my arrival, and had discussed a joint project connecting IS, medicine, and economics.

279 I had explored the topic in a recent class report.
Appendix A: The SoftHeart Story

physicians. Early proposals reflected these ideas, employing an experimental approach to test the framing and multimedia effects on patient outcomes (Aug. 17, 1992 and Aug. 26, 1992).\textsuperscript{280} It appeared the project was underway.

\section*{9.2. Health Funders Skeptical}

In January 1993, the health agencies responded to our proposals. Although some of the referees supported our claim, others had serious doubt about information framing and a computer’s ability to influence patient behavior and biochemical outcomes. The onus was on us to prove IT’s systematic effect on health.\textsuperscript{281}

Funding rejections forced us to more fully consider the role of the health and patient context in moderating and perhaps negatively influencing the outcome. What was the role of the provider? Had Clinic One been closely consulted about the intervention, and would the clinicians follow strict guidelines for discussing and presenting the information? What about the literature that suggested the widely varying effects of information on different patient populations? New measures were introduced to measure the path from the information intervention toward patient attitudes, behavior, environment, and finally biochemical outcomes to provide process measures of impact.

\textsuperscript{280} At this point, very few people at the clinic had been fully consulted about the project. In fact, I had mistakenly believed that the physician colleague of my supervisor was the clinic director, which later turned out to be incorrect.

\textsuperscript{281} The results of this first round of funding illustrated the widespread disagreement amongst health providers about the benefits and disadvantages of using computer systems. Some believe it leads to improved decision making and patient treatment, while others suggest computers destroy rapport between the provider and the patient.
9.3. Our Response

Responding to the criticism throughout the remainder of 1993, we developed new proposals reflecting a move away from information framing, toward the use of more extensive IS and IT to provide a strong test of IT’s effect on patient outcomes. If the agencies felt the treatment was too weak, then we would try a larger IS intervention to provide a substantial test of the information effect. In addition, Clinic One’s director and personnel were consulted more closely to observe their work and initial reaction and support of the project. The proposals also carefully explained the term ‘information system’, encompassing patient/provider communication as an essential part of a clinical information system.

At the same time, proposals were being altered to reduce the cost of technology required to carry out the projects. Some of the proposals included only a single computer to collect and print various reports for both the physician and the patient. Therefore, the proposals emphasized the effect of information and information provision on provider/patient communication, and the method of producing this information was considered irrelevant.

---

282 A full analysis of clinic one’s information system and the feasibility of alternative IT and IS was conducted in April of 1993 by me and three MBA students.

283 Although the second set of proposals included more technology such as patient computerized questionnaires, we soon discovered that the health agencies were also unconvinced about the computer’s effect on patient behavior and outcomes.

284 I once thought that the information could have been provided by carrier pigeons or a team of graphic designers in the back office producing charts for patients and physicians. If health agencies were unconvinced about the use of computers, we could use any method to produce and deliver information to test its effect on patient outcomes.
During this period, I began to study the heart disease and health promotion literature to understand more fully the causal chain between information technology and patient outcomes. The literature did suggest that the IT-patient outcome chain was filled with many detours and possible traps that required addressing in our design. In addition, presentations of the proposal to medical groups also suggested that the planned intervention would produce too much change in the organization and creating future implementation problems. New proposals were submitted on February 28, August 30, and Oct. 4, 1993 reflecting these changes and views.

9.4. Old and New Environmental Shocks

During late summer of 1993, the government announced that hospital one was closing for cost cutting measures. Initially shocked that the project might suddenly die, I received word from the director that Clinic One and other outpatient clinics would remain at hospital one. However, despite government promises, Clinic One was moved to hospital two in early 1994, and the project was pushed to the background as the organization struggled with the move and the new setting.

In addition, old problems continued to haunt the project. Health agencies continued to reject the proposals. The complexity of the project’s IS intervention, which was the result of the criticisms in early proposals, created other doubts about the revised

285 This was particularly influenced by my first meeting with my future health promotion advisor on November 23, 1993.

286 A presentation on March 4, 1994 before several researchers and physicians not connected with the clinic commented that the change was very large and would involve a massive effort in the clinic.

287 Clinic one was moved into a temporary facility in an old nurses residence, using very small consult rooms located one wing away from the administrative office.
IS’s ability to exclusively target the experimental group without “contaminating” the control group. Agencies continued to be concerned about the cost of the computers, and the lack of tested support for a system that had not already been constructed. It was a “catch 22”: we needed the money to build the system, but the money would not be provided until the system was built.

While trying to revise the proposal to satisfy health agencies, the research proposal had now crossed another line unacceptable to a different group. During my proposal defense on March 22 1994, a number of researchers within the MIS department at the my university asked the question, “Why is this MIS research”? By dropping computers from the intervention to deal with the health groups’ concerns about cost, I was trying provide an inexpensive method of testing improved information and presentation on patient outcomes and deal with the concerns of cost from the health agencies. At my thesis proposal defense, a number of IS researchers now considered the technological intervention too simple, and were dissatisfied with my focus on meeting the demands of the health setting, and not the demands of advanced IS development and testing.

9.5. Decision to Move Ahead

Fearing that time was short, I began to search for IS design ideas that would satisfy both the MIS and health funding agencies throughout the middle of 1994. I started to develop simple programming add-ons to the information already contained in Clinic One’s EasySys software, using my own time and money. To do this, I needed access and user involvement from the clinic personnel to determine how best to access and present
the information. Specifically, I targeted physicians, the director, and the coordinator in order to get an understanding of EasySys, and to gather suggestions and commitment to an intrusive change to the clinic’s information system.

The EasySys software used by Clinic One, provided a number of opportunities for system development. My experience teaching and working with dBase IV and my general knowledge of database design allowed me to dissect the software’s architecture. Written in Clipper, a dBase programming dialect, it had been used for many years to collect patient, lab, dietary, and medication information. However, its reporting facilities were limited, its data extraction capabilities were slow and limited, and its source code was not available for modification.

Originally, I began using Microsoft Access to test the data connection and presentation capabilities. My previous experience on a project using Microsoft Access provided me with this expertise. However, Access proved to be too slow because it attached itself to DBF files through a translation process. DBF files were not native to the program.

As a result, I searched through software packages that used DBF files directly. I chose Foxpro for Windows in May of 1994, a dBase dialect that appeared to link directly with the old clinic files, to provide a tool for extending data reporting. Foxpro for Windows also claimed to have a software package that could graph data, a necessary tool.

---

288 The ‘database file’ (DBF) format is a standard data format for many database applications.
for achieving many of the reporting goals. In this respect, I was trying to leverage and capitalize on a resource already available at the clinic.

The next task was to learn Foxpro programming. Like most programming languages, many of Foxpro’s commands were far removed from the language of the setting and what was wanted from the software. More importantly, it was difficult to determine the general approach to programming in Foxpro. To assist me with this task, I purchased a book that provided important concepts for programming clean, efficient, and maintainable code in Foxpro (Goley 1993). To this day, the structure of the program has been heavily influenced by the conventions outlined in the book.

The difficulty of converting design ideas into programming code, however, was still high. For example, a “browse” window constructed in December of 1995 in scheduling provides read only, edit, and calculated fields for date, time, class name, spots allowed, spots available, and day of the class. Each of these fields has validation checks to ensure that the time and date entered is valid. In addition, it marks with an asterisk those classes of the target patient, and marks with an @ those classes worked by the target staff member. Each time the filters are changed on the side, the schedule window needs to be refreshed with those classes that meet the new filter criteria. Each time the user moves to a new record, the staff list in the edit windows is updated. And finally, the browse

---

289 This graphing using MS Graph turned out to be incredibly slow (Sep. 17 1994). As a result, I purchased an external graphing program on October 6, which reduced the time for graphing from 30 seconds to 3. The graphing module has formed the backbone of the project.

290 The use of PRG (program) files that call SPR (screen program) files created a modular and maintainable design. In addition these PRG files contain the functions called by events in the SPR file, because the book warned against the embedding of code in the objects. It claimed the code was difficult to maintain when embedded in the objects. We will see that this became a problem in the second case study and the use of object oriented programming.
window also locks and unlocks the record during editing to allow multi-user access and change to the classes. This code is also supported by additional functions (e.g. f_unlock) that are run when certain events occur. To reach this understanding of the task, context, and programming language took one year.291

The following is the code for this function:

```
BROWSE FIELDS patmark=IIF(f_patinclass(timessch.tnumb),"**","");H="":1;W=F.;V=f_unlock();
stfmark=IIF(f_stfinclass(timessch.tnumb),"@"," ");H="":1;W=F.;
timessch.tdate :H = "Date":W=f_unlock();
l_day = LEFT(CDOW(timessch.tdate),2) :R :H="Day";
timessch.tstart :P="@R 09:09" :H = "Start*":W=f_whenget(VARREAD())
 :V=f_valid("START");
timessch.tend :P="@R 99:99" :H = "End":W=f_whenget(VARREAD())
 :V=f_valid("END");
timessch.tdescr :H = "Descr":V=f_unlock();
timessch.tmaxpats :H="Spots":V=f_unlock();
l_spots = (timessch.tmaxpats - timessch.tnumbpats) :R :H = "Open”;
WHEN f_showinfo(timessch.tnumb);
FOR f_browsefor();
WINDOW w_classes;
IN WINDOW w_sched;
TITLE "Classes";
NOWAIT;
NOCLEAR;
NODELETE;
NORGRID
```

With the onus on me to develop something of interest to the clinic, health groups, and IS groups, I began developing prototypes by attaching the new program to the old data, and trying various display and reporting formats. The purpose was to learn what could be done in the Foxpro program while at the same time produce a working version of software.

291 There are new software programs that have appeared since the project began, such as Visual Foxpro and Microsoft Access, that claim many of these language issues are solved by their programs. But given the complexity and specific needs of each context, I am doubtful whether higher level programming language will solve the problem of transforming the problems and opportunities of a context into the software (Brooks 1987).
Using EasySys' data files also required effort to understand and interpret the files and fields in the database, and how the data reflected the nature of the clinical setting (Aug. 12 and Aug. 17 1994). The risk calculator was the main focus, because I felt it would provide the ‘value added’ to the organization and produce information different from what was already collected. The graphing of data also provided another avenue for value-added information presentation. I also began developing a statistical calculator that would explore the significance of correlation between various variables for an individual patient. All three were considered ‘derived’ data from EasySys.

9.6. Prototype Demonstration Sparks Interest and Constraint

The presentation of a prototype on September 29th, 1994 began a new period in the project, as users and groups became interested in the software. Directors and physicians were excited by the possibilities. The demonstration sparked interest in the capabilities of the program, including the graphing as a teaching tool, the use of a risk calculator to summarize large amounts of information, and the potential to expand the database to collect additional information for research purposes. Because the system was aimed at the physicians and their adoption, the enthusiasm was good news.

---

292 At this point, the software was quietly called the “adherence application”, reflecting the health promotion research question about information’s effect on adherence.

293 Many people at the meeting knew about me, but did not have a clear understanding of the research goals until that meeting. I had been working in the background for over 2 years at this point, and it was only now that people were learning about the project.
The program also sparked a number of comments and concerns, suggesting that more user involvement was required. These comments indicated diverging approaches and interests within the organization. For example, physicians disagreed on the use of total cholesterol (High Density and Low Density Lipids) in risk calculation, instead of singling out Low Density Lipids (LDL); the bad cholesterol. Two, the use of Systolic Blood Pressure (contraction) instead of Diastolic Blood Pressure (relaxation) in the risk calculation raised other eyebrows. Recent literature, explained one physician, suggested that the relaxed blood pressure (diastolic) was a better predictor of heart disease. Three, the automatic generation of risk reports using natural language text produced some major concerns about physician liability and the ability to edit. Four, the Framingham data is based on a population that has previously not had a heart attack, while a number of the patients at Clinic One were “rehab” patients recovering from a heart attack. Finally, the system was not addressing provider administrative problems such as consult letter generation, mailing, and other administrative areas.

I also discovered the research group at the prototype demonstration in September 1994. I learned quickly that they were very interested in the project, and their vision of an integrated network linked by telephone connections began to add support and vision to the networking aspect that had been ignored thus far by the project.

The first meeting with the group, however, revealed strong demands on SoftHeart for strict data entry, a need for medication modules, strict documentation standards, and a

294 The calculator was based on well established epidemiological data from the Framingham study that had examined thousands of individuals since the 1950s.
strong belief in OS/2 as the computer and network operating system (October 5). First, I felt there was an emerging issue of ownership of the software, which to that point had been designed and programmed by me. Second, I felt the group was attaching itself to the intellectual and future ownership of a software product that had been solely my work to that point. Finally, my association with the group fractured a fragile relationship with the first coordinator of Clinic One.

9.7. Conflict Erupts

Unknown to me, there had been an on-going conflict between the coordinator and the research group based on a software program developed four years ago by the research group to replace EasySys (CholeSys, a pseudonym). CholeSys's interface and design looked very different from the EasySys, and had strong validity checks of data that required the clicking of buttons in order to clear the screen. The purpose of CholeSys was not for administrative ease of use, but for correct data entry for research and patient information integrity.

EasySys, on the other hand, had been the sole development project of the coordinator. Using clinic funds, she had worked with a programmer to produce a system that reflected her specific wishes. These characteristics included easy data entry, character fields to enter comments and asterisks to highlight data, key punching navigation through the program, etc.

295 The designer and creator of CholeSys worked for the research director, and was a former lab employee. Lab systems are designed to store the name of the individual entering data for every item entered into the system. This provides for responsibility and validity checks on data input.
CholeSys, having been written in dBase and transported into Foxpro after encountering software problems in dBase, had several software bugs. When a report from an ‘independent’ external consultant criticized the program’s operation and maintenance, the CholeSys project was halted. Comments by the research group and the coordinator indicated a chronic conflict.\(^{296}\) When the research group associated and joined my project in mid October, the coordinator must have viewed this association as evidence of resurrecting CholeSys.

This entire conflict may not have erupted if my project had not seriously affected EasySys’ operation. In fact, to ease the programming required, this was a preferred strategy. However, technological constraints with EasySys created barriers to achieving improved information reporting. First, being a stand-alone software package, EasySys locked the database files, preventing other programs from accessing the data while it was in use. Second, every piece of data was stored as a character string.\(^{297}\) As a result, calculations from the data and graphing required extensive use of functions to convert the character strings correctly. Third, important fields required for complete risk calculation using the Framingham tables were missing in the program. Finally, the coordinator was either unwilling or unable to get access to the original source code from the programmer.

As a result, I began to examine EasySys more closely, and started to build EasySys’s functionality into my software (Feb. 9, 1995), using look-and-feel...

\(^{296}\) The research group commented on poor data entry in EasySys and the coordinator’s killing of a good software product (Oct. 23). When discussing the past with the coordinator, comments included “are those guys still trying to push that system”? (Oct. 17).

\(^{297}\) Even calculated values were computed using a calculator and typed into the system.
characteristics from both programs. This request seemed reasonable because the coordinator had mentioned earlier that Clinic One had planned to move EasySys into Windows, and an integrated package addressing both input and output would be ideal.

The result of these intrusions turned the coordinator’s from a hesitant and lukewarm supporter into an unfriendly foe. At the time, I felt the coordinator’s reaction was justifiable given the previous work and time spent developing EasySys and maintaining clinic operations. I also assumed that the coordinator knew the job, and should comment on any part of the project that would adversely affect it. Therefore, I accepted her comments to design the data input interface similar to EasySys.

On February 23, 1995, I decided to give a presentation of the Lipid Program prototype to the coordinator. I wanted the coordinator to have some input into any changes that may be desired, and to show how I was progressing. The demonstration of the emerging software in Windows went very poorly. Being partially completed, the coordinator criticized the software for missing functions and any slight differences in navigation, including bugs in the navigation in the older program that I had fixed. The coordinator commented that I should not demonstrate the program until it was done, and to stop wasting time. When asked what could be improved in this new version from the Lipid Program, the coordinator stated a fax number could be added to the physician record, and typing months using characters instead of numbers would be preferable.

---

298 On February 23, 1995, insider reports from one employee suggested that the coordinator did not enjoy my presence, and mentioned that I was causing too much change for the coordinator.

299 This seemed very incremental to me at the time.
At the same time, the coordinator pointed out various barriers that would influence my project, including a new hospital wide system that may affect the project, and a new ‘information’ ethics form developed by Clinic One. Further questioning about these barriers did not reveal what and how these would affect the project. On February 28, the coordinator gave me the new ‘information’ ethics form, that required researchers and investigators to provide project information before they had access to the clinical data. It surprised me that I had been working on this project for over 2 1/2 years, including direct work on the database files developing this new software, and I would need permission to work with EasySys’ information in the future.

9.8. Decision to Act

Concerns about the coordinator’s support for the project, the lack of agreement amongst clinic personnel on key design issues, controversy about the use of treating lipid disorders and medications (Feb. 14 1995), the lack of time and commitment from physicians (Feb. 9), the loss of my laptop in a home robbery (Feb. 8), and a funding rejection (March 27) threatened the project’s survival. I also received an offer to work on another project that appeared to guarantee usage data by its completion (March 21).

My first step was to draft a list of conditions toward my continued involvement in the project (March 21). The conditions included a focus on clinical and light input

---

300 At the time, the coordinator stated that she was mentioning these barriers to ensure the success of the project.

301 A talk with the hospital’s ethics lawyer revealed that the coordinator had developed the form within the last few weeks (Feb 22, 1995).

302 The clinic was unable to provide me with any computing resources to work on the program. As a result, I had been using my supervisor’s laptop for the last 6 months. In addition, the computer also had patient information on the drives. Luckily, no one could get into the software without knowing a great deal about programming.
modules to keep the programming time to a minimum, a suitable start date for implementation (June 1995), equal control with the coordinator, director and myself over any administrative changes, equal ownership of the source code for modification and future sales, and 1/2 hour per week with each physician to discuss the system. A meeting was arranged with my two committee members and the physician colleague who had allowed us access to his clinic in the summer of 1992. I mentioned the alternative software project. He commented that the software was a gift to the clinic, and he would not let the project die. He described the coordinator as a secretary, employee of the clinic who should not be determining policy. Unfortunately, he said the lack of leadership at the clinic was jeopardizing the project’s success.

I then had a meeting with the director of Clinic One on March 22 to discuss my conditions. He said that the coordinator has problems with other people stepping on her territory. He felt that the system failing would be a tragedy after so much work, and would do what was necessary to make it work. He promised that the clinic would be moving into the new offices by end of July/August, that he would try his best to get funding for the computer hardware by that time, and that he would meet with the coordinator.

The director did meet the coordinator later on that week (March 29). The director asked the coordinator to get the source code for EasySys, but the coordinator stated it was impossible, and that she supported the position of not allowing access to the source code.

---

303 During the presentation, the coordinator was asked by the director about the source code and she responded with sadness and defiance. I believe that the inability of the director to control the coordinator in front of his physician colleagues was an embarrassment and prompted his aggressive manner in dismissing her.
code. Her explanation was that at least part of the software program was built on top of another program that was marketed several years earlier, and was proprietary. The director was angered by her response, and when the coordinator asked if the clinic would be better without her, he said yes.

On March 29, I appeared before the entire clinic staff, presented my software and the list of conditions. One physician mentioned that I was being courted by another project, and how the clinic wanted to keep this project. Talk about the benefits of the system toward clinical operations, the potential to earn money selling the software and the 50/50 split in project royalties dominated the presentation and conversation. The physicians met behind closed doors after the meeting, and decided to fire the coordinator.

9.9. Project Builds Momentum

There were many events during this period after the firing of the coordinator that pointed to unlimited growth and development in the project. Many people believed that with the removal of the coordinator, the project was free of restrictive constraints (Mar 31 1995). On April 8, I began restructuring a copy of EasySys' database, with the intention of replacing EasySys with SoftHeart in the future.

---

304 The firing came so swiftly and unexpectedly. A number of personnel mentioned that I was not to blame for the firing. Conflict between the coordinator, a few physicians, and other clinic personnel had been occurring for a number of years.

305 Character fields that contained lab data were converted into numeric fields, thus closing the door on EasySys' future operation in the clinic.
At the same time as conflict was brewing between myself, the coordinator and the research group, other groups within the hospital showed an interest in the project. Discussions with the research group suggested a vision of HeartNet linking outreach clinics with Clinic One (Oct. 4 1994). Clinic Two was mentioned on Nov. 10, 1994 with our meeting with the hospital’s MIS group. The merger between the two clinics was discussed, and by the accounts of personnel in both clinics, SoftHeart appeared to be an important component in uniting the two clinics into a single program (Nov. 10 1994).

The hospital’s MIS department became interested in the project, both for competitive and cooperative reasons (Nov. 10 1994). Throughout the period from November 1994 to March 1995, I began to understand more about the IS structure within the hospital, including various systems that appeared as possible replacements or complements to SoftHeart. These included the ADT system (Administrative Discharge Tracking System) and PCA (Patient Clinical Administrative) (April 3), CAREVISION (a clinical system to handle patient data) (May 3), and a system in lab medicine.  

On March 30, the same day as the meeting with Clinic One personnel that was followed by firing of the coordinator, I met the coordinator and the director of Clinic Two. I immediately attracted to the clinic because of its focus on health promotion (exercise, stress reduction, smoking reduction), the integrated and cooperative relations between the staff under the direction of a charismatic cardiologist, and its limited use of computers. Clinic Two appeared to be an ideal organization to develop and implement

---

306 However, I had currently been unable to see these systems and understand their infrastructure to determine whether they were a threat or benefit to the project.
SoftHeart. A demonstration of the software on April 11 seemed to impress the director of Clinic Two. He commented that SoftHeart would require only a few changes to suit their needs. Given their need for computerized systems to report patient statistics to Hospital Two, however, he would be satisfied with just basic demographics for now. My goal was to produce a limited version of SoftHeart for immediate implementation.

Efforts were also being made to obtain funding for the project through various sources. A talk on April 6 with the director of Clinic One revealed that $20,000 with the hospital foundation was earmarked for prevention. On April 11, the director of research and I began work on a proposal for money to build the LipidLink infrastructure.

April 13 also began my own movement of the software into administrative areas that had been fully controlled by previous coordinator. That day, patients at Clinic One had been waiting two to three hours to see the physician. Discussions with the head clerk revealed that scheduling and booking was “way too manual”. I felt that the software could address both issues, and that the construction of a scheduling system would be ideal (April 25).

There was also an opportunity to hire a programmer for the summer, based on an offer by the director of Clinic One. Although initially hesitant to take on the extra load of coordinating the work with a second programmer, I decided that it could be worth the time and effort. I met with the student on May 6th, and decided to hire him on a 2

---

307 One of the alternatives that my committee and I considered was leaving clinic one and switching to the second clinic.

308 The programmer was a friend of the director's son.
week trial basis at the beginning of June. From all perspectives, it appeared that the project had momentum, resources, and unlimited freedom.

9.10. Development Quagmire

Despite the coordinator leaving, and the perception that the development would now continue at my own pace and to a conclusion, I began feeling the endless refinement and development that was continuing to overrun the project (Apr. 25 1995). Throughout the period I continually felt exhaustion and worry about the future of the project.

I began to realize that a “blank cheque” on development had opened the doors to a software “sinkhole”, especially with the research group who were now able to press demands without interference from the coordinator (May 3 95). Expectations of a quick implementation with Clinic Two were soon dashed with a prototype demonstration to the coordinator of Clinic Two and a major list of demands to change the software (May 4 95). Refining the database to include the unique scheduling and structure of Clinic Two also required major changes to the database structure (May 26; June 24). Data could no longer be tied directly to every visit, as patients with Clinic Two came many times with only three major data collection points. It took until July 24 to figure out the best database structure to handle these differences.

The ever extending implementation date for the new program combined with a never-ending list of requirements being generated by the clinic threatened the delay and completion of my thesis project. The start date had been pushed to June (Jan 24), increased to July 17 (Mar 30), pushed ahead to August 17 (April 27), later moved to
Given that I had been working on software development for over 1 year and two months by that point, with various earlier dates in the year proposed as starting dates, frustration had begun to set in.

Software design and development continued to drain huge amounts of time and resources, despite some initial breakthroughs. A switch from a purely display and presentation program to data input prompted dramatic changes in orientation and software code (Apr. 25). In addition, all the functions converting character data from the EasySys software files needed to be removed and replaced with direct access to the new numeric fields (Apr. 11).

Combined with the dramatic software restructuring was the endless list of requirements that continued to pile in from many opposing directions. The backlog included requests for building field level security to record the name of the person entering each field into the database, a need for radical changes to the database structure, getting the physicians to understand, support and eventually use the system, setting up and testing new hardware, integrating Foxpro for Windows into the OS/2 platform, training and listening to clinic personnel, building better medication and physical exam catalogues, and addressing documentation and maintenance concerns.

The promised resources and support for the project were nowhere to be found. Furthermore, the early summer weather seemed to cause clinic personnel to lose interest. The student programmer who arrived in June was not provided with a computer and office

---

309 The actual starting dates did not occur until early 1996. Clinic one moved in early January, clinic two in early February, and the first usage of SoftHeart began in early April 1996.
space at the hospital during the first couple of weeks, requiring me to house him at the university with my computer (May 26; May 29). By June 20, three weeks after beginning, the programmer had not yet been paid. Attempts throughout June to get the programmer's wages were met with indifference and forgetfulness by one of the research group's personnel. Finally, after continual phone calls, the programmer was paid on July 6th, six weeks after beginning work.³¹⁰

Loss of key personnel who supported, helped develop, and/or provided key information about the organization and the functions of SoftHeart also affected the project's course. The two coordinators from each clinic, and several clerks from the first clinic left during the course of the project. The second coordinator had a health promotion background, and understood the theories and modules in SoftHeart that could address areas not currently covered by the clinics. One of the clerks in Clinic One had been a key source of information about the clinic and a sympathetic supporter of the project (Aug. 22). A second clerk provided important input into the scheduling and patient modules, and would have been a supportive user of the system during testing and debugging. All three resigned during the summer and early fall, forcing us to reconsider and develop relationships with other and/or new individuals to continue development and support implementation.

In order to develop the system with more input from users, the programmer and I moved from the university to the hospital on June 27. We were greeted, however, with

³¹⁰ Talks with the research director suggested that while the director of clinic one hired the programmer, the research group had never received any funds to pay for his wages. The lack of coordination and communication was evident.
general apathy and lack of time by clinic personnel, suggestions that my thesis would never be finished because of lagging development, and computer crashes.

Perhaps the most serious and frustrating shocks were the computer crashes during late June and July (June 28; July 10; July 26; July 27), and the finger pointing and hostility it created between the software group (myself and the summer programmer) and the hardware group (research group). Dealing with OS/2 and new computers purchased by the research group, it was initially difficult to determine the cause of the computer crashes. Being skeptical about the research group’s support and use of OS/2 over Windows as the operating system for SoftHeart, I originally attributed the crashes to OS/2. On the other hand, the research group placed blame on me, my external backpack hard drive, and the programmer working with me that summer. The testing of my backpack drive by the hardware company supplying the research group apparently caused a computer crash. However, I was unable to talk directly with them about the cause. I reached the conclusion that the research group and the hardware group were protecting their reputations by blaming me.

There were several other conditions that prompted me to consider leaving. Stress was a big factor in my visit to emergency on July 30th after falling down and injuring my head. A trip to Washington on August 10 to learn the software used by Empower indicated that funding, resources, and consideration for my needs were better on the

---

311 This would lead to a continuing anxiety between the research group’s network technician and myself for the next several months.

312 The external backpack hard drive attaches to the printer port of a computer, operating as an independent hard drive.

313 I am a diabetic, and stress causes instability in blood-sugar control.
Appendix A: The SoftHeart Story

Empower project. Third, while trying to fix a computer as a favor to a clerk on August 15th, the director of research sarcastically commented that the computer has likely contracted a virus from one of my diskettes.\textsuperscript{314} Discussion with my committee about my treatment and the offer to join the Empower project from my one advisor prompted me once again to consider switching projects.

I reached the conclusion soon afterward that facing spiraling software development time, the lack of input and interest from the clinic personnel, the time and effort expended in fixing and dealing with the virus, an offer to work on the Empower project, and pressure from my thesis committee, I indicated to the clinic that I was leaving the project. On Aug. 30, I reached the conclusion that apathy had killed the project.\textsuperscript{315}

9.11. Recovery and Slow Growth Toward Early Implementation

A meeting with the physician colleague of my supervisor on August 27th began a series of surprised reactions by many personnel in both Clinic One and two. He was surprised that I was leaving, and surprised at some of the causes of my leaving. He had no idea of the struggles with the software, unlimited demands, and problems with the research group. He suggested I send a letter to the clinics outlining what was happening.

\textsuperscript{314} There was something about the comment that pushed me too far, and resulted in another letter and meeting with my committee.

\textsuperscript{315} During the same period, the summer programmer and I worked furiously to get a completed module together for the two clinics that would at least replace the old lipid software (Aug. 19; Aug. 21; Aug. 25). Being completed, I felt that I was leaving with a limited product close to completion. Since I had hardly received any money for the entire work, I felt only somewhat guilty for leaving.
I was surprised to see that many people at the clinic wanted me to stay. The coordinator from Clinic Two took me out for lunch. The new coordinator in Clinic One saw the system as an important component in her future work. A talk with the director of Clinic One yielded surprise and concern over my leaving. The director of the research group was also surprised and confused that I was upset with his group. He asked, “are you mad at the machinery or at us?”

Once it was discovered that it was the Monkey II virus\footnote{The Monkey II virus is a boot sector virus released in early 1994, that infects the boot sector of a disk. The virus spreads by activation of a floppy disk in the floppy drive, and copies itself onto the hard drive of a computer. This virus then operates in memory, affecting other fresh floppy disks inserted into the computer. The virus eventually destroys the boot sector of the hard drive making it impossible to access the drive.} that caused the computer crashes, the project began to settle down.\footnote{However, future crashes of clinic computers continued for the next 6 months as other floppy diskettes from other sources infected the machines. It wasn’t until the IBM anti-virus package was installed on every machine that the occurrences of the crashes decreased.} Discussions with the clinic groups reduced my time with the software to one day per week, requiring the clinic to work more efficiently with my time, and to take responsibility for further software development (Sep. 7). I also joined the EMPOWER project officially, thus providing me with an alternative case site to study software development and usage. Therefore, despite continued problems with commitment to the project from software testing to eventual usage (Oct. 11; Oct. 13), the Empower project’s release valve stabilized my need for usage data.

The research group also took a strong lead on the project, as my role moved to the background as a consultant and software advisor. This leadership has been important in achieving the viability of SoftHeart away from its designer. The original focus of SoftHeart, however, continues to change from clinical and patient care to research on
large datasets. First, the lack of commitment and resources toward computers and reporting to physicians in consult rooms points to this change in emphasis. Second, the focus on implementation in the administrative areas has pointed to the data entry personnel responsible for supporting the research database. And finally, work on capturing admitting information and lab information automatically from the hospital systems is trying to reduce the reliance on internal data entry, which is prone to clerical errors and constraints.

General meetings with the clinic personnel indicated the clinic had revitalized its search for project funding, and was interested in implementing the system soon; bugs and all (Sep. 28). One physician mentioned channeling $50,000 from the outreach program into SoftHeart's development because the outreach clinics in the province would be interested in using the software.

The server finally arrived on October 17, providing solid evidence that the project was going to move ahead. At the same time as these constraints were placed on the project, the clinics received $30,000 from the hospital foundation, and there was discussion about another $100,000 coming from a drug company interested in the project (Nov. 2, 1995).

---

318 This movement towards a research focus was later assured by funding of the HeartNet and SoftHeart project from a drug company interested in drug research.

319 Continually, I was reminded of how the physicians would be unwilling to use the computer in the consult rooms; especially data entry.

320 $80,000 was received from the company two weeks later.
The clinic also was considering hiring a programmer based on my recommendation (Dec. 2 1995). This would greatly reduced the need for my own programming. In addition, despite the initial training period, it meant that I would eventually be able to leave the project by releasing the clinics' dependency on my expertise.

Talks turned to methods of implementation (Jan 8, 96; Feb. 14), and a move to the new clinic location occurred in early January. SoftHeart went through an early implementation effort, working with clerical staff to begin the transition from EasySys to SoftHeart. Some interesting surprises emerged during on-the-job training and readjustment of the software to deal with work issues. Gaining an initial foothold in a busy and chaotic environment by April 9th has been challenging and frustrating at times (Jan. 30). Initial bugs and missing functionality required by the administrative group prompted early cries for dumping the program and sticking with EasySys. In addition, the need to integrate the system with the wishes of the hospital’s MIS and admitting groups has been difficult both for the software and the clinic human resources required (Jan 31 96; Feb. 28 96) during implementation.

However, quick response times and relatively easy identification of software problems by myself and the programmer provided assurance to users that the software can work and will be supported (Mar 6; Mar 7). The results so far have been encouraging. Many personnel at the clinic are now referring to SoftHeart as essential for future survival.

---

321 This programmer was the same programmer who I was working with on the Empower project.
322 The new area looked very impressive, and was fully wired with fibre optic cable for a computer network.
Appendix A: The SoftHeart Story

against funding cuts. SoftHeart appears to be becoming essential for allowing new opportunities in clinical, administrative, and research tasks.

9.12. What Lies Ahead

SoftHeart is far from being completed, and despite its early success, still has a number of known and unknown hurdles to overcome before it becomes ‘infused’ and ‘institutionalized’ into the clinic. External threats to its survival from the hospital’s MIS and lab medicine groups still remain, and the technical stability of both SoftHeart and HeartNet have not fully been attained. Technically and resource-wise, however, the project has the capability to thrive.

Additionally, there is the question of SoftHeart’s focus on either research, administration or clinical care. This issue has been central in understanding the reactions to the prototype in September 1994, the conflict over EasySys during March, the software explosion in late summer of 1995, and the implementation in April 1996. SoftHeart’s focus began with a clear direction toward patient care. Using the same software, SoftHeart is now being implemented and further developed towards research and administration.323

---

323 This may be a step in the wrong direction, may simply automate and increase computing power in areas further removed from patient care, and may produce little impact on patient outcomes.
10. Appendix B: Events in the Development of EMPOWER

This appendix reports on the various events during the development of Empower, providing the raw data for Chapter 5. These events will be discussed within various process stages of the project. The process stages include project emergence, getting ready, Pandora's box opens, squeezing the most from Toolbook and EMPOWER, and visions of a new software constrained by resources.\(^{324}\)

10.1. Project Emergence

My first exposure to the Empower program and the project director was at a meeting on Nov. 23 1993.\(^{325}\) The director seemed interested in my work at the two clinics, and I was interested in what I called the ‘means-ends’ chain model of health promotion planning that was used in Empower for planning breast cancer interventions. Having an interest and need to learn more about the health promotion field, I began to get involved in various projects with the project director.

During mid January 1994, I was called by the project director to discuss a copyright issue with the Empower software. Having worked recently on a copyright study, I said I could help. I learned from the project director that the EMPOWER software had been

\(^{324}\) Note that these stages, although presented sequentially, were overlapping to some degree.

\(^{325}\) I originally thought the software was called ENABLER.
developed by a US company, with the director and two colleagues serving as advisors. The director was interested in building a Canadian version of the software, and was negotiating with the US company on the division of royalties from any future sales of derivative versions. Questioning the director, I found that he and his two colleagues had been paid for the work they had provided, and thus would have to pay royalties to the company from any derivative products produced in Canada.

In early February, I met with the director and a sales representatives from statistics Canada to discuss various methods of obtaining data from Statistics Canada's on-line service. The purpose was to explore methods of supplying specific epidemiological data to users of Empower without requiring us to collect and store the data. I also began taking a course with the project director about his planning theory and model in January of 1994. During the summer, I was involved in a workshop focusing on the planning model and the Empower software. In addition, I began working on a project with the director involving the collection of information for another project involving heart health, and the development of an information system to assist in the collection and measurement of community health programs.

---

326 The one colleague would later join the US company as a vice president in charger of a division that would develop and market EMPOWER and other information technology products.

327 At the time, we were focusing on redeveloping Empower for heart disease prevention and planning, because a proposal on this topic was before a heart disease agency.

328 I helped monitor the computer labs.

329 This project involved the collection of data from community heart health projects to measure the outcomes of the interventions. I used accounting principles in a software design to integrate and measure the various events in communities. Perhaps because of its abstract nature, the software was rejected by our co-investigators with the provincial government.
During 1994, I was unaware that various proposals for funding on topics of tobacco control, heart and stroke prevention, mammography screening, and self care had been submitted to selected agencies for funding upgrade and expansion of the Empower software for the upcoming year.

Early in 1995, in anticipation of a tobacco control grant being funded, meetings took place to discuss modifications to the software for tobacco control. The meetings on February 28th and March 2nd produced more questions than answers about the proposed new software. Important user and setting characteristics that might be the target of the software, including the geography or location of the planner and/or target group, the disease area, the target population, the setting (worksite, clinical, or school), the users (planners, consultants, community members), and the goals of the users (planning, expert suggestions, design, etc.), were identified at the meetings.

Afterward, I spent several hours afterward trying to collect my thoughts about the purpose of the software, and devising a strategy for determining how the EMPOWER 'black box' works. I needed to learn more about the Empower program, which had occupied a large portion of the next discussion, held on March 6. I worked with the software over four days and identified various components and concepts used in Toolbook, the multimedia authoring tool used to produce Empower.330

Hopes were soon dashed temporarily on April 3 when the tobacco control proposal was turned down by the funding agency. The project seemed in doubt and confusion to me

---

330 I felt good to be working on a project that was moving forward given some of the frustrations I was having with the other two clinics at this point.
over the next two weeks, until the meeting on April 26th to discuss the mammography grant.\textsuperscript{331} This project had been funded by the National Cancer Institute of Canada (NCIC), and discussion about the organization of the project was beginning. An offer to work full-time on the project as the designer and programmer was turned down by me in mid-May 1995. I was continuing to work with the two clinics, but I volunteered to devote one day per week to the project as a software supervisor and designer. It appeared that the project was underway.

\textbf{10.2. Project Direction, Decision and Organization}

On March 2, 1995, still uncertain of the task that lay ahead, a series of issues for the project were discussed at a group meeting. These issues addressed many different levels and aspects of the project. For example, freedom from copyright with the US company, navigational freedom in the software, data access using the software, the purpose of test workshops (rapid design and testing or input for later testing and diffusion), and future modifications to the software for other settings and disease areas were some of the issues and goals considered.\textsuperscript{332}

As early as March 6, 1995, the group was discussing a division of labor between the “systems” or “structural” team and the “expert knowledge” or “content” team to match user expertise with specific tasks. The “content” changes were to drive the development effort by identifying changes to the wording, functionality, and text resources. Any changes that could

\textsuperscript{331} I was very confused at this point in time. I did not understand why we were still meeting if the proposal had been turned down recently.

\textsuperscript{332} Integrating these various goals with specific tools and software development plans was difficult and complex.
be completed without programming, were also to be carried out by the content team. On the other hand, revisions that required "structural" changes to the program were directed toward the programmer and me.\footnote{\textsuperscript{333}}

The most important decision facing the group was whether to modify the US version of the software or to begin a new version from scratch. This 'make or buy' decision was not a discrete event, but has been continuously revisited throughout the project. Various discussions around this topic emerged early (Feb. 13 95), as I began raising the idea of an alternative platforms, such as Foxpro, to effectively manage the data produced by Empower. On March 6, I presented my current understanding of the Empower software, with some reservations about buying the US software. First, code was embedded in a multitude of objects throughout screen design panels. Second, the setting of the software, Canada, was different from the US version. Finally, because we were considering tobacco control at the time, the content of the new software would be very different. I recommended, however, that we should stick with Toolbook for now, and make modifications and upgrades.\footnote{\textsuperscript{334}}

Some of the issues about the make-vs.-buy decision emerged throughout the summer. Advantages of modifying the US version appeared to include a nearly completed product with beta tested source code (thereby reducing the need for extensive user involvement), the reduced use of development resources, and support from the US company. Advantages of

\footnote{\textsuperscript{333} We also started using the language of Toolbook, EMPOWER, and databases to describe the our goals and programming changes. Toolbook language included 'books' and 'modules', 'pages', 'scripts' of code, and 'system' books (Mar. 4 1995). EMPOWER language included vision statements, missions, health objectives, behavioral and environmental objectives, factors and factor objectives, strategies, actions, and activities. Database talk revolved around the storage and access of data, and included database files (DBF), memo fields, character fields, records, indices, updates, searching, etc. This terminology was deliberately created based on the suggestions of another IS researcher on the team.}

\footnote{\textsuperscript{334} The "for now" allowed us to accept the US version conditionally, subject to further evidence.}
starting from scratch included freedom from the US copyright, avoiding the difficulty of learning the code, developing the software with our vision, building a faster program, decreasing the size of the program\(^{335}\), addressing the fragmented storage of information, and overcoming some undesirable interface characteristics of the US version\(^{336}\).

Continuing to explore the Empower and Toolbook software, one of the key pieces essential and missing in the evaluation was the documentation of and/or discussions with the designers of the US version. Actions to get this documentation produced only a few pieces of information (May 17)\(^{337}\).

On June 17, 1995, the team did get a chance to meet with the designer and the other health promotion researcher who worked with the project director on the US version. The meeting was also a chance to begin clearly defining the project’s focus and direction, and the research questions to be studied in the two communities. These issues were to a moderate extent determined by our analysis of the US version and its needed revisions.

The meeting revealed a number of important findings. First, the project director stated the original intention behind the US Empower was to provide a structured planning tool that ensures users plan from the top (population-based), and avoid planning from the bottom (resource based). The result is a focused health intervention that addresses important concerns of the target population\(^{338}\). Second, the software was a compromise between the educational

\(^{335}\) Currently, EMPOWER requires over 12 Megabytes of disk space.

\(^{336}\) These poor interface characteristics included page-by-page navigation, lack of standardized screen appearance, and the large number (96) of pages.

\(^{337}\) We learned subsequently that the programmer for the US company had never provided the documentation of code, and after leaving the company, could not produce it.

\(^{338}\) Although I agree generally with this perspective, the software enforced this view through strict navigation of every page, without the ability to move back and forth. This navigational restriction would prove to be a problem in the community in January 1996.
Appendix B - The EMPOWER Story

orientation of one of the health promotion researchers, and the planning focus of the project director.\textsuperscript{339} Third, the focus of this project is to “Canadianize” the US version of the software, including the development of a French-Canadian version. Canadianization appeared to include mostly content changes to the screens and database files.\textsuperscript{340} Fourth, there was little evidence that anyone was using the US version regularly for planning mammography screening interventions. Fifth, Empower’s database capabilities were limited, and the method of changing project and resource data files appeared difficult.\textsuperscript{341} Sixth, the designer of the US version, a close colleague of the project director, was concerned about the complexity of the program and the ripple effects of changes to the software.\textsuperscript{342} Finally, the designer was only able to spend a few minutes showing the code and program to me, and I was unable to resolve many of the concerns I had about the software and its suitability for modification and dissemination.

Throughout July, I continued to examine the software to determine where information, display and input screens were located. The search revealed several problems with the software. First, it appeared that many methods were used to store similar types of information, making it difficult to understand and explain the structure to other team members. Second, most of the data were stored inside books, book fields (July 12), and

\textsuperscript{339} The project director mentioned the “consult-on-tap” versus “consult-on-top” approach to providing information to users. Through negotiation, it was decided to use the consult-on-tap approach. However, other educational pages of information are presented to users when entering all six books.

\textsuperscript{340} Assuming that the US version was bug-free and relevant to breast cancer, I mentioned to the group that we would not require a programmer if Canadianization required only content changes.

\textsuperscript{341} Again, I raised the possibility of using Foxpro as either a replacement or as an add-on to the program to handle data retrieval and storage. At the time, I was worried that I may be applying my Foxpro experiences from the two clinics into a project where it may not be appropriate. I was beginning to see the world’s problems with database solutions.

\textsuperscript{342} Previous to this statement, the designer had spoken harshly to the group about the copyright issue, and that modifications to the program required royalties to the US company.
compiled help files, making it difficult to evaluate and modify information. Finally, the location of summary report rules that provided expert advice to the users began to concern me (July 12). I was hoping that the information could be exported out of Empower into database files that could be changed more easily by database software, and then imported back into the program.

Despite these reservations, my trip to Seattle for a course by the producers of Toolbook soothed many of my fears (Aug. 10 1995). In the comfortable environment of the training center, I learned about the great advantages of Toolbook, and also discovered that the company had recently released a Database Connection program that could potentially address the data storage problems with Toolbook. After the course, I produced a handout for the group claiming that Toolbook was not as limited as I had originally thought (Aug. 12). Strict revision of the US version of Empower would also put us closer to the December deadline for the revised version. Choosing to redevelop and extend the US version, the development team and I believed that the US version was working correctly, and that our group was only adding additional functions and changes to a working version.

Having chosen to modify the US version for the Canadian setting, other project pieces came together. A search committee's advertising and interviewing yielded a promising programmer (Aug. 28; Sep. 6). Version control and documentation had been discussed and was being established (Aug. 12; Sep. 10). Documentation procedures were discussed

343 Later in September, we found that the rules were hard coded using 'if-then' statements in the books, making evaluation and change very difficult.

344 The goal was to produce a working version by the end of each week, ensuring that time was spent on focused projects that could be completed within the week. Backups of the data were to be stored on new floppy disks every Friday, and I was to take a copy of the program home in order to have two physically separate locations for the program.
The project appeared on the move, now with adequate financial and human resources.

10.3. Pandora's Box Opens

During user testing and exploration of the software, the programmer and I began to see many design and programming problems. Spurred and blinded by the possibility of quick success, the team continued to hope the software would yield to programming breakthroughs. Instead, the project began to sink into a programming quagmire.

Without documentation, the programmer and I were faced with the difficulty of exploring code attached to each object in the US version of EMPOWER, including push buttons, fields, pages, backgrounds, books, and four large system books (Sep. 14, 1995). This code was very frequently undocumented except for brief comments, and a significant portion of this code was possibly outdated and no longer being used. This made it extremely difficult to review and change functionality. As a result of unclear documentation, changes to the program often had unexpected and unwanted results. On many occasions, a single change turned into a string of changes required to fix rippling bugs. By mid-October, we learned that there would be no additional documentation forthcoming from the US company. The team was left to read “OpenScript” code to determine how the program worked.

This task was made even more difficult by the object-oriented programming language of Toolbook. Each object in the book contains its own separate code encapsulated in

---

345 Documentation was to allow me or someone knowledgeable of programming to step into the project should anything happen to the programmer.

346 OpenScript is the programming language in Toolbook. Described as friendly and “English-like”, the many commands and object referencing capabilities made interpretation difficult.
“handlers”. Handlers are activated when certain events occur, such as a mouse clicking the object, or moving the object, etc. If an event occurs and an object does not contain a handler, the program searches up an object hierarchy to the page, to the page background, to the book, and then to ‘system’ books, searching for a handler with software code. In addition, calls to other objects’ handlers activated automatically by events in the current object could occur. As a result of this implicit activation of software code and little documentation to understand the basic approach of the programmer in designing the code, changes to the program did ripple in unexpected ways.

Almost immediately after the beginning of the project, our initial assumptions about Canadianizing, deAmericanizing, and adding to a tested product proved incorrect. A large list of software bugs, navigational traps, strange or incomplete summary reports, and unusual screen layout problems were identified at a project launching meeting on September 8th. Substantial content problems and changes were suggested. In addition, visions of new and improved functionality were also added to a growing list of requirements. It was hoped that many of the changes that required database storage and retrieval would be obtainable using the new database connection software from the Toolbook supplier.

Perhaps the most serious outcome of this sudden and expanding wish list was that the software was directing our development efforts. The program was filled with many screens

---

347 For example, an enterPage handler in an object would run its code if the page is selected by a user.

348 This list of changes was so extensive, that as of March 31, 1996, many have been abandoned.

349 These new functions required a lot more time in formulating the requirement, assessing where it would fit into the program, and then building and testing the completed code.

350 As an example, the deletion of factors affecting behavior was a desired option unavailable in the US version; once a factor was selected it could not be deleted. For planning and modification of plans, this was unacceptable.
that required fixing, and the fixing also led the group into exploring new functionality. The project was losing its focus. As a result, the uncertainty with the software and how the expanded list could be incorporated into the software, and the lack of early success and timely completion of assigned tasks, all increased the stress level of both the “content” and “structural” teams. Losing sight of a core “vision” for the software revision and development, the change lists reflected a reactive approach to the software, where problems and functions on each of the 96 pages drove development. Being rather loosely constructed, the US version led us in many directions fixing functions that in retrospect may not have been essential to health promotion planning.

In terms of content, immediate problems were raised. First, case examples were repetitive (Sep. 8, 1995). Second, with the software’s focus on mammography screening, other important breast health practices such as breast self examination (BSE) and clinical breast examination (CBE) were being ignored (Sep. 8, 1995). Since three behaviors and one environmental factor were hard-coded into the screen and software, change without major modification was impossible. Third, the strategies and action steps were behaving erratically in module 5, an important module linking the factors affecting the three behaviors and one environmental element to specific action plans. Four, certain screens were suggesting that the age 40+ group of women could be a target population, despite research suggesting

531 Case examples, “responding here”, and “why” text provided information to the users about the specific page they were viewing.

532 These three behaviors included women’s mammography seeking behavior, and physician examination and recommendation for screening. The one environmental factor was the availability of mammography screening machines.
the that women between 40-50 were at a substantially lower risk. Finally, the US health objectives for breast cancer were hard-coded into the program, thus requiring a search for similar Canadian objectives and the need to determine the structural changes required to incorporate Canadian objectives.

On October 10, 1995, a new and unexpected problem emerged. The printing of Empower version 1.1’s database files containing resource information on strategies and action steps revealed four different versions. After two weeks of comparing the versions, it appeared that each of the four had missing data. The decision was made to print all four and try to produce a fifth with the good information from each of the three, and additional information from recent literature searches.

Key to our development approach of the US version was to outsource the data management functions into another program. However, as early as October 3, 1995, early tests opening these files using dBase IV and Foxpro caused structural changes to the files rendering them unreadable back in Empower. Believing that the files may contain a memo file from the older dBase products, dBase III+ was used, but again it converted the file so that it

---

353 This 40-50 group is perhaps the most controversial topic in the breast cancer community today. Those who claim systematic screening of this group is a waste of resources are counteracted by others who say their risk is high enough to warrant screening.

354 This was discovered after anomalies in module 5 indicated that the program was selecting strategies and action steps incorrectly. The strategies and action steps provide specific interventions for important factors affecting women’s behavior, provider recommending or performing breast exams, or availability of mammography machines.

355 The question of obtaining a master version was discussed as early as August 15, 1995. However, the question was never fully addressed until the content problems emerged.

356 The database files in Empower used the same DBF file structure as dBase and Foxpro. We assumed that using Foxpro or dBase would allow us access to the files for changing and updating information.
was unreadable in Empower (October 10, 1995). Our reconstruction strategy appeared to be failing.\textsuperscript{357}

At the October 23rd meeting, I also raised concerns about the multiple methods of storage in the Empower program. Identical flat file databases were sometimes stored in a book with repeating record fields or in a database file (DBF) using a dynamic link library.\textsuperscript{358} Sometimes, information was stored in properties of the book or in database files. Finally, behavioral and environmental factors affecting mammography screening were hard coded directly into the program’s screen design, while factors affecting those behaviors and environment were stored in a database file.\textsuperscript{359}

As we learned more about the software, we realized that many of the problems in the US version were due to the mismatch between the desired database functionality and the choosing of the Toolbook software platform. To allow several users to use the same books and at the same time to capture specific information for concurrently running projects, an unusual method of storing the data in database files (DBF) using a cumbersome dynamic link library was being employed. An EMPOWER database file was created where each record represented a specific login and project name. The record was composed of numerous memo fields relating to each page or category of information. These memo fields stored free text

\textsuperscript{357} As late as May 2, 1996, a perceived breakthrough with a software package still produced distorted information files. At that point, we have cut off any further development on strategies and action steps updates.

\textsuperscript{358} Database file updating, deletion, and insertion is handled easily and effortlessly by most common database applications. However, use of the dynamic link library required detailed coding prone to errors. The database connection package was to prove even more slow and bugged when tested in the future.

\textsuperscript{359} It appeared that the discovery of a code library to link data with DBF files during the development of the US version had been introduced and used to store any new data, while older data continued to be stored in books or book properties.
and text representing rows of numbers separated by commas. Extraction and interpretation of this information was extremely slow and difficult.

In addition, objectives for each specific project were stored in another database file called OBJECTIVE, where each record represented an objective tagged with the project login and name. Searching for objectives to fill book information required a sequential search of the database by project login, name, and objective name. In addition, it was discovered that deletion from the database file was not possible, and troubles emerged in updating the information.

Finally, problems with content and simple regurgitated information in the summary reports, the only real output of the entire program, raised serious concerns. Finding the location of the rules was important to fixing and improving the summary reports (June 17; July 12; Sep. 18, 1995). To our surprise, most of the advice was printed regardless of user input, most of the information was simply user input, and a large amount of user input was not being printed (Sep. 25). To our further surprise, we found the rules hard coded in two system books (Nov. 20). “If-then” statements surrounded each text statement by grabbing information from certain pages and checking to determine if the condition held.

Throughout October, I provided mixed messages about the software that may have confused the issues. On October 23, when asked if we were on target, I mentioned that the project was on schedule except for the summary reports. This comment was based on my continuing belief that we were making enhancements to an already complete software product. However, earlier in the month (October 10), I had stated that the software currently

---

360 Summary reports use expert rules to insert information based on user responses.
does not operate functionally as a planning tool. I told the group I was concerned with overall design, and that the product was poor for storage and retrieval of plan data; a crucial area. I criticized the US company for the poor product.\footnote{There may have been many reasons for the software’s poor design. One of these reasons may be the innovative and unknown nature of the product at the beginning of project conception. Our team could critique the software’s problems, realizing that the US version allowed us to avoid these problems. The US company also had the two consultants who were experts in the model and content, but not in the programming problems they were creating by making frequent late-stage requests for changes in design.}

Choosing to extend the US version of EMPOWER, both the programmer and I had felt that any problems with current and future functionality could be handled by outsourcing those parts to the database connection, other executable programs, and/or external program libraries.\footnote{Particularly, the dynamic link libraries (DLL) available in a windows environment.} Each outsourcing strategy, however, proved to be very problematic. A graphics program library was linked into Toolbook with great difficulty (Nov. 16). The Toolbook database connection program, which was crucial in our decision to stick with the US version and Toolbook, was later abandoned after testing and use indicated the code libraries were slow and filled with software bugs (Nov. 9 - Dec. 8, 1995).

The complexity of the software and the uncertainty regarding its problems and capabilities created a number of problems in communication and coordination between the “structural” and “content” arms of the team. Bottlenecks in the content area created idle programming time because the health promotion researchers were swamped by the amount of work required in assessing the entire program, and as other projects and demands took up their time (Nov. 6, 1995). The difficulty of integrating the team into a common understanding of health promotion and programming concepts created problems in understanding requirements and programming constraints and options. This problem was exacerbated by the
research assistant’s and the programmer’s lack of knowledge about each other’s areas, the
research assistant being a health promotion planner but new to computers, and the
programmer’s knowledge of computing but new to health promotion planning. This language
connection problem made it difficult to explain programming problems and options, and kept
the programmer and me from making ourselves understood regarding why certain
expectations could not be met.

Because of the lack of documentation from the US company, our own system of
documentation proved difficult to establish and enforce, and the programmer found himself
more interested in coding than dealing with details. This interest in coding was also increased
and reinforced by pressure and rewards to continue programming the numerous changes to
the software. Also, given 96 pages of screens and code embedded in many objects, the
complexity of the program reflected itself in the complexity of our documentation.

By November, a series of conclusions and questions faced the project. One, the news
of an incomplete module (Sep. 24; Nov. 7) revealed to us that the software had obviously
been patched together without full testing (Nov. 20). Two, and as a result of the time already
taken in fixing the software, the French translation of the software was in doubt (Nov. 7).

Three, with the designer of the US version seriously ill, and his chief programmer no longer
employed by the US company, it also appeared that very little would be coming from the US
company to assist the project (Nov. 16). Four, I raised the need for the project to develop a
general strategy, and to identify what the software should do (Nov. 16). Too many new ideas
for changes were being raised, and it was becoming difficult to determine priorities. Five, we

363 We had planned fixing the English version before sending the program for French translation.
began to assess what other "software" would be required by Community One to work with the program. A discussion ensued about the software manual, training, user knowledge of the planning model, etc., that would be required to sustain use of the software.

On November 24, I began using Foxpro to build a contact manager that I hoped to eventually attach into the Empower software. Problems with the database connection continued. At a meeting on December 11, 1995, I mentioned that the database connection program was not working, and the contact manager could not be redeveloped and improved through the package. Faced with a decision to outsource the functionality to Foxpro, C++, or continue with the Toolbook version, we decided to stick with Toolbook. With a January 23 start date for Community One, and our programmer moving to half time by January, we had little time to explore other options. Suggestions were made to disable features that were bugged or had data irrelevant to the Canadian setting (Nov. 24, 1995; Dec. 2, 1995). In addition, plans were made to ensure that the earlier modules were ready, and later modules would be worked on during January and delivered to the community once available.

Meanwhile, thoughts turned toward redevelopment.

10.4. Squeezing the Most from Toolbook and EMPOWER

Despite the problems emerging with the software, both the content and the structural teams began to work within the constraints to achieve a number of changes to the software.

The earliest changes began during July and September, as the content team began working on the problems with the page resources: case examples, 'responding here', and

---

364 Using Foxpro would have required a fixed investment of 2 Megs for the code library and another 100K for the program itself. Given the already large size of Empower, we chose to work with the Toolbook version.
Appendix B - The EMPOWER Story

'why' information. American case examples, repetitive information, nonsensical statements, and poor English were fixed to more appropriately fit our anticipated users and the Canadian or other non-US contexts. In addition, language and layout changes to the pages were conducted at the same time, to eliminate poor English and simplify interface appearance.

Perhaps one of the earliest difficulties and desired changes to the US version was the strict navigation requiring page-by-page movement (Mar 2; Sep. 7, 1995). To improve navigation, individual module map pages showing the structure of the module pages were constructed, and direct access to these pages was made available from the general map page. In addition, the general map page, which used to be a series of buttons to each of the six books, was replaced with a drawing of the planning model (Dec. 2 1995; Jan 29 1996). Clicking on different parts of the model including the model background provides access into each of the books. Navigation was also improved by reducing the number of screens and dialogue boxes. For example, a number of resource questions requiring many screens of data entry in module one was replaced with a spreadsheet page, allowing easier access, improved interface design, and quick data entry (Oct. 23). As another example, button bars in Module 6, once contained on separate panels, are now placed on a single panel (Nov. 14). Finally, push buttons to access action agenda reports and the summary reports were provided to allow easier access to important information (Oct. 16).

365 Since this information could be modified using Empower itself, this information was changed directly by the content team.

366 However, user testing in community one still revealed problems with the page-by-page navigation once past the map pages, and the need to complete the six modules in order.
Despite problems with the database connection, the team was able to outsource a number of functions to Microsoft Write (Oct. 23), and a graphics package for graphing time data (Nov. 16, 1995). However, graphing of epidemiological data was canceled despite early success because of data storage problems through the database connection.\textsuperscript{367}

Changes in the research on mammography screening have been collected but currently not added to the strategies, action steps, and factors affecting the three behavioral and one environmental factors affecting mammography screening. In addition, updates to the reference list, university and non-university sources of information and support for mammography, and general intervention strategies have not been carried out because of time constraints.\textsuperscript{368}

Most of the software bugs have been removed from the program, thus providing a minimum of technical feasibility to the software. Summary reports are reporting more of the information recorded by the user, and more suggestions are being provided. In addition, the reports are linked to MS Write automatically, allowing the user to change and save the information into a word processor.

\textbf{10.5. Visions of a New Software Constrained by Resources}

Throughout the project, alternative methods of outsourcing, augmenting or replacing Toolbook as the development platform have been discussed (Feb. 13, 1995; Feb. 28; Mar 2;
During late December 1995 and early January 1996, discussions about producing a new software to overcome many of the problems in the current EMPOWER version were beginning to emerge. Perhaps the strongest criticism of the software came in January and February 1996, just before the introduction of the software into the BC community, when I criticized the software as unusable and inappropriate for supporting planning (Jan 15, 1996; February 26). This prompted an initial round of activity and suggestions for developing the new version of EMPOWER. Design ideas and documents were distributed and worked with the group throughout the period until March 1996. In addition, resources were used to buy “platform” software for the project in March.370

In addition, I described the concept of a “means-ends” chain that would support a new ‘generic’ Empower. Data would be stored centrally and drawn upon as a resource (Jan 22, 1996). The entire health promotion model was to be seen as a means-ends chain linking higher level goals of quality of life, to lower level goals such as health, behavior, factors affecting behavior and environment, and interventions (Jan 16, 1996).371 The program would be dedicated towards allowing users to build factors and linkages between factors for a target population. The purpose was to simplify the six module approach into a set of simple

369 Alternatives included Foxpro, C++, and Delphi (a suggestion from the programmer).

370 Resources included Delphi (Pascal) with database capabilities, a few educational versions of Visual Foxpro, textbooks on software design and engineering for the group, InfoModeler for modeling the database structure and properties of the new system, and Windows 95. In addition, I supplied the group with Visual Basic programs including a word processor, spreadsheet, image editor, and graphics drawing program.

371 My vision of Empower as a means-ends chain extended back to my first meeting with the project director on November 23, 1993. Throughout 1995, I continued to think about the means-ends chain as a method of simplifying the Empower concept, and as a method of storing data (Sep. 8, 1995).
constructs. This would provide a software construct for guiding development (the chain), and provide a more open tool for use. In fact, most of the development would not involve programming, but the construction of factors and linkage for a specific area. For example, a heart disease version would include behavioral and environmental factors connected to heart disease, and factors predisposing, reinforcing, and enabling these behaviors and environments.

Problems that emerged in Community One with the contact list and linear nature of the software also seemed to provide an impetus to move to a different software platform (Feb. 19, 1996). Especially serious was an emergency meeting of the development team to deal with trapped data and poor functionality with Community One's contact list on March 8, 1996. Given my comments and discussions with the US designer, the project director suggested that the software be scrapped or used as only an educational version. A discussion of alternative funds to support the development effort arose (March 18). The end of March was spent searching for software resources to build a new platform (March 19).\(^{372}\)

However, environmental constraints including the funding agency's expectations of a finished product and the expectations of waiting users and communities has forced us to continue working on the program in Toolbook. Second, there is also the uncertainty of a new and unknown software product versus a known and developed software.\(^{373}\) Third, the overlap between the different versions of Empower and a concern as to how each was to be funded and tied together forced us to refocus on the current version (Feb. 18). Finally, some positive

\(^{372}\) Eventually, we have chosen to use Borland's Delphi product, with the paradox engine attached to support database retrieval and storage. The lower level language, and its ability to link easily with Visual basic controls (VBXs), OCXs and dynamic link libraries appears promising.

\(^{373}\) The project director recently stated "a bird in the hand is worth two in the bush". It is also true that the two birds in the bushes will never be caught if we never attempt the catch.
impacts of the software on the planning process in Community One also suggested that the software may be having an impact despite the problems (Feb. 22). Meanwhile, demands for changes to the old software continued to pile up (March 27, 1996). On March 28, 1996, the director of the project stated that changes to the old Empower were to be the focus until June, when a major workshop with the software was to finish.\textsuperscript{374}

EMPOWER has an interesting and uncertain future. A version of EMPOWER will most likely succeed, but it is difficult to determine which version and how that version will be used to educate and support planning. Project timelines are also difficult to predict.

However, the research promotion group has secured both me and the programmer as a continuing resource on the project, and future development and dissemination of the EMPOWER software in many disease and behavioral areas is planned.

\textsuperscript{374} Also, writing this thesis prevented me from providing enough time on the project to supervise development.
11. Appendix C: Contextual Information for Case One - SoftHeart

11.1. Introduction

The search for health funding and the need to understand the context of heart disease and the health care organization prompted an extensive literature search. Important areas of interest were heart disease epidemiology, theories of health behavior, the health context, and the health funding debate. Each provides my current and continuing understanding of the various medical and administrative influences on SoftHeart.

11.2. Heart Disease Epidemiology

My continuing understanding of the epidemiology of heart disease has been important to software development in the two heart clinics of case one. The two heart clinics in case one are involved in the treatment and prevention of cardiovascular disease. This section elaborates the terminology and epidemiology behind cardiovascular disease (CD).

Cardiovascular disease is a general term for diseases affecting the cardiovascular system. The cardiovascular system involves the heart, lungs, head, liver, and kidneys. Two systems carry oxygenated and deoxygenated blood; the circulator system and the pulmonary system. The circulator system carries oxygenated blood through the coronary arteries to the different organs (except the lungs), and drains deoxygenated blood. The pulmonary system carries deoxygenated blood to the lungs for transfer of carbon dioxide and new oxygen
Blockage of the blood vessels or inadequate flow (heart failure) are the initial stages of any CD disease. Any failure or blockage in the cardiovascular system will lead to one or more of the following:

1. Myocardial Infarction - death of a heart muscle from a blockage.
2. Angina Pectoris - severe pain in the chest from exercise because the coronary arteries cannot meet the blood demand of heart.
4. Heart Failure - the heart stops pumping or dangerously reduces its blood flow.

What Causes CD?: Most risk factors in health are highly inter-related and causality is difficult to prove. No one factor seems to be the starting point for the other risk factors, the cause of intermediate effects on the cardiovascular system, or the final causes of CD. Keep this in mind when reading the risk factors of CD.

The following is a classification of risk factors for CD (Kaplan & Stamler 1983):

(I) Host Susceptibility:

Host susceptibility are unmodifiable patient characterizes that correlate with CD:

1. Genetic and family history - A severed condition called hypercholesterolemia causes high cholesterol levels that are not influenced greatly by dietary changes (Hayden 1988, pg. 9). Medication treatment is usually prescribed.
2. Age and Sex - older men are more susceptible to heart disease than younger females.
Estrogen levels in women seem to act as a protector against heart disease until women reach menopause.375

(2) Clinical Conditions

The following clinical factors are indicators of CD risk, and are largely dependent on intermediate dependent variables leading to heart disease. Most of the clinical conditions are caused by the more independent living habits to be discussed later:

1. Blood lipids or cholesterol levels are highly correlated with death from coronary heart disease. High levels of Low density Lipids (LDLs) in the blood cause atherosclerosis; a disease that causes fatty plaque to build on the wall of the arteries (Stamler 1992). This may lead to blockage of the arteries (Kaplan & Stamler 1983, p5), or to hardened arteries that cannot expand to accommodate increase blood demands (Sarafino 1990, p 60).

2. Blood pressure also contributes to atherosclerosis and is also a sign of thickened arteries (Kaplan & Stamler 1983, p 9; Stamler 1992).376

3. Diabetes causes high hyperglycemia (elevated levels of blood sugars) contributing to atherosclerosis. Once gain, diabetes is possibly an indicator of poor diet, as well as a contributory factors in heart disease.

(3) Living Habits:

375 This female/male dichotomy is somewhat controversial. The rates of heart disease in women appear to be rising as women have begun to smoke and involve themselves in more stressful environments.

376 A local pharmacist recently advertised a blood pressure monitor available in their stores for routinely checking blood pressure. Consistently high results were to be reported to a physician.
Cardiovascular disease (CD) is somewhat preventable through medication and patient changes in lifestyle; diet, alcohol consumption, exercise, and smoking. The following sections describe living habits that are considered more modifiable by the individual, and generally more independent and controllable than the other risk factors:

1. Smoking causes blood pressure to increase, exercise capability to decrease, oxygen needs to increase, and HDL cholesterol levels to drop (HDL removes harmful LDLs from the blood). According to some studies, those who quit smoking even after many years have only half the risk of CD as those who continue to smoke (Kaplan & Stamler 1983, p.9; Stamler 1992). Risk of death from CD is directly related to amount of smoking.

2. Obesity appears to be more a symptom or intervening variable leading to heart disease, caused by lack of exercise, improper diet, and/or genetic factors. Obese persons have twice the risk of cardiac failure and brain infarction.

3. Physical exercise studies suggest that among 4 exercise categories, the lowest exercise category had twice the risk of cardiovascular mortality as the most intensive exercise category (Kaplan & Stamler 1983, p 11).

4. Psychosocial risk involves stress and overwork. When these are accompanied with a lack of control over the pace or conditions of work, they are highly correlated with CVD. Related to stress is a personality syndrome referred to as Type A, one dimension of which (hostility) is associated with type A individuals. Once again, the effect of stress and its correlation with depression, diet, smoking, exercise, and other
risk factors makes it difficult to isolate its specific effect (Theorell 1992).

(5) Diets high in saturated fats, sugars, sodium, and reduced fiber contribute to atherosclerosis and the lowering of HDL cholesterol (Kaplan & Stamler 1983, p 11, Stamler 1992)

(6) Excessive alcohol intake increases hypertension and damages the myocardium (rhythm of the heart).

Treatment of cholesterol and heart disease is controversial (Moore 1989; Kuller 1992). Especially controversial is the use of lipid lowering medications that produce unpleasant and potentially harmful side effects, and there continues to be a question of whether cholesterol is a symptom or a cause of heart disease (Windeler & Gefeller 1992).

Understanding heart disease epidemiology is important for system design in case study one. Important factors contributing to heart disease include host susceptibility, clinical indicators, and living habits. Controversy of the association between cholesterol and heart disease, and the treatment of cholesterol especially using medications, has been an important influence on the system development for the first case study.

11.3. Theories of Health Behavior

Having examined the health care environment and health planning, there is a need to understand the theory linking various factors of the PRECEDE/PROCEED model together. Research examining these connections include theories of health behavior (predisposing, enabling, reinforcing, behavior and environment) and epidemiology (behavior and environment affecting health). It has been important in both cases to understand the various
ideas and theories addressing individual behavior, and the role of the health provider and the health planner. This section explores theories of health behavior. The remaining two sections explore the epidemiology of heart disease and breast cancer.

A number of theories and ideas explore the factors affecting the health behavior of a target population. Many of the theories arise out of work in social psychology, marketing, sociology and community development. There are both intrapersonal theories examining the individual more closely, and interpersonal theories focusing more on the individual's environment and its influence on patient attitudes and behavior.

A number of intrapersonal theories predict adherence\(^{377}\) to a medical or health "regimen"; a set of behavior or behavior changes to achieve good health. These theories include the health belief model, theory of reasoned action, multiattribute utility theory, attribution theory, and consumer information processing. Intrapersontal theories focus on the beliefs, attitudes, and information processing capabilities of individual patients in predicting adherence.

The health beliefs model (Rosenstock 1990) has three important constructs predicting adherence. Three attitudes about disease are measured including susceptibility and severity, outcome expectations of treatment including perceived benefits and barriers, and self efficacy including the person's perceived ability to perform the treatment.

Theory of reasoned action and multiattribute utility theory focus on the individual's adherence intentions by measuring attitudes toward a specific regimen components (e.g.

\(^{377}\) "Adherence" is defined as the closeness of an individual's behavior to a desired behavior prescribed by a health provider. Another term for adherence, used mainly in the medical literature, is "compliance".
reducing smoking by 5 cigarettes) (Carter 1990). Two constructs emerge from reasoned action. *Attitude* including the likelihood of health outcomes performing the behavior and the evaluation of those outcomes. *Subjective norms* include the patient’s beliefs about what others believe they should do, and the patient’s motivation to comply with these norms. Every important outcome and each treatment regimen component needs to be identified and presented to patients to predict behavioral intention of adherence. The theory suggests that information should be used to increase the value of outcomes, and the perceived likelihood that treatment regimen will produce these outcomes.

Attribution theory focuses on an individual’s beliefs about the disease and their locus of control over these causes (Lewis & Daltroy 1990). Similar to the concept of self-efficacy found in the Health Belief Model and Social Learning Theory, patients who attribute disease to factors that are controllable, stable, and internal will have better adherence and health outcomes. Also, attributing successes to internal, controllable, global and stable factors can increase adherence.

Consumer information processing (Rudd & Glanz 199), a final intrapersonal theory, states that consumers are limited information processors, and require information presented in format-friendly and “processable” form in order to deal with these limitations, and to reduce patient search costs. More importantly, the simple provision of additional information and repetition increases learning and recall; a necessary condition for adherence. As well, physician communication problems and lack of time with the patient also affects the amount of information that can be provided, and problems with technical jargon reduce the ability to deal with limited information processing.
To summarize, intrapersonal theories predict patient adherence by assessing individuals' beliefs toward disease and treatment regimen, and by improvements in information provision to deal with poor information processing. Information can improve adherence for forming and changing attitudes and attribution, and by increasing recall and learning through improved information clarity and repetition.

Interpersonal theories examine the social support and environmental factors that regulate and influence patient behavior directly through attitudes, or indirectly through non-cognitive routes. The purpose of these theories is to explain why intrapersonal theories are sometimes weak at predicting adherence, and to augment more encompassing theories. Intrapersonal theories have been especially weak in explaining adherence to chronic diseases such as diabetes, renal dialysis, and other conditions in which on-going behavioral maintenance requires social reinforcement and sometimes direct support. Information plays an extended role by influencing and educating social support members, and by providing extended appraisal and emotional information to patients. Interpersonal theories include social learning theory, social support theory, and theories about patient-provider interaction.

Social learning theory (Perry, Baranowski, & Parcel 1990) focuses on the environment, and the “situation” defined as the individual’s perception of that environment. Constructs in the theory include expectations of outcomes, expectancies (values of these outcomes), behavioral capabilities with self control leading to performance, reinforcements of behavior, observational learning, self-efficacy, and emotional arousal.

Social support theory (Israel & Shurman 1990) focuses on stress, coping, and problem solving and its relationship with social support and control. Stress is related to adherence in
that individuals experiencing severe stress with the disease and the treatment regimen will reduce adherence. Social support assists in alleviating stress and enhancing control through emotional, instrumental, informational, and appraisal support. Informational and emotional support includes advice, suggestions and information about the disease, and treatment regimen. As well, information provided to spouse and other health providers can produce additional emotional and appraisal support. The results of improved social support can be increased feelings of control and self-efficacy towards the treatment regimen

Patient provider interaction (Dimatteo & DiNicola 1982; Joos & Hickman 1990) suggests that patient provider interaction is poor, resulting in a lack of patient adherence and dissatisfaction. Physicians have difficulty interviewing patients about sensitive topics, attitudes, and concerns. Patients leave the clinic with a poor and incomplete understanding of the treatment regimen because of jargon, differences in physician/patient background knowledge, and physicians’ poor understanding of patient information needs. This lack of understanding about patient perceptions and attribution reflects the disparity of the patients’ mental models and the medical view of disease. Asking patients about their views of disease, severity, causes of disease, desired regimen and/or concern with regimen may assist physician diagnosis and explanation of treatment regimen. As well, increasing patient expression of concerns and needs, and allowing patient suggestions for treatment increase adherence.

Social influence, a theory regarding the sources of physician power in patient/provider interaction, suggests that informational and referent power are the most important tools of the physician, because they encourage internalization of the treatment regimen (Kelman 1958).
Research confirms this view by showing that providing more information to the patient increase adherence; especially if it provides patients with an ability to participate in choosing and assessing the treatment regimen (Rodin & Janis 1982).

To summarize, interpersonal theories explain and predict adherence to a treatment regimen by focusing on social and environmental factors, such as the actions and attitudes of social support members, to explain nonadherence with individuals who are fully informed and hold initially positive attitudes. These factors influence adherence either directly through attitudes, or indirectly by non-cognitive routes such as reinforcement.

Various theories and approaches to promoting healthful behavior are used to guide researchers and practitioners. These include both intrapersonal and interpersonal theories of health behavior. These ideas have shaped the development process in the case studies.

**11.4. The Health Care Context**

This section examines the language, areas, and funding mechanisms of Canadian health care. Understanding the health care context has been important in developing a system that can address environmental groups and hospital constraints and opportunities.

The goal of health care is to prevent disease-related death (mortality), and disability (morbidity). The means to achieving these ends are *efficacious* and *effective* treatments. *Efficacy* means the treatment works under ideal, usually experimental, conditions. *Effectiveness* means the treatment will work in the real medical world, where patient adherence and physician prescription and description are less than ideal.

There are two treatment approaches and funded areas in health care to address
mortality and morbidity: preventive and curative medicine. Preventive medicine\textsuperscript{378} (PM) attempts to prevent disease in patients before the disease occurs, or to detect disease early enough to prevent much of its potential harm. Preventive medicine can be classified as either primary, secondary, or tertiary. Primary prevention tries to influence well people to lead a healthy lifestyle, without necessarily targeting groups with a high risk of disease. Secondary prevention finds those individuals at high risk of disease or who have signs or symptoms of disease, and uses methods of detection and early intervention. And finally, tertiary prevention occurs after a disease has manifested, and tries to reduce its impact and rehabilitate the patient. The treatment in preventive medicine include changes in the working/living environment of the patient, and changes in their risk behavior (Sutherland & Fulton 1990, p 162).

Curative medicine (CM)\textsuperscript{379} deals with disease in the final stages when the symptoms and effects of the disease appear. Treatment involves surgery, drugs, or both.

In 1990, health care consumed 9\% of the GNP in Canada, as compared with 7\% in 1970 (Evans, 1990). Of this expenditure, 70\% are funded by the provincial governments (Royal Commission Report, 1991, p A-11) with support from federal transfer payments, and have consumed between 31\% and 33\% of provincial government expenditures since 1980 (Ibid., p A13). The total amount spent by the province of BC in 1992 was $5.4 billion (Vancouver Sun February 15, 1992).\textsuperscript{380}

\textsuperscript{378} Similar terms include clinical health promotion, disease prevention, behavioral medicine, etc.
\textsuperscript{379} Sometimes called acute care.
\textsuperscript{380} $2.54 billion of this money went to hospitals, $1.4 billion to the physicians and fee-for-service, $570 to continuing care institutions, $389 million to community and family health, $286 million to a provincial drug plan to pay individuals for
Over the past decade, the health care environment in Canada has generally been chaotic as governments struggle with shrinking budgets, a slowly growing economy, and dissatisfied tax payers. There have also been wide variations in medical practices inside and outside Canada (Rachlis & Kushner 1994, pp. 91-94), wide differences in health outcomes amongst regions within the same province (Seaton et al., 1991, p B-6), and criticisms of physician prescription of unnecessary treatment (Rugush 1987; Rachlis & Kushner 1994, pp. 91-92). From this environment has emerged several proposed funding and delivery approaches to health care to solve this apparent crisis.\textsuperscript{381}

Current funding of hospitals and physicians employs a cost budgeting approach. Hospitals are funded with annual global budgets, and physicians are reimbursed on a fee-for-service basis.

Most health regions/hospitals receive funding in a lump sum every year through “global budgeting”. It is then the responsibility of the administrative branch of the hospital to allocate these resources to the various departments in the hospital including emergency, acute care, diagnostic services (radiology and lab medicine), and outpatient care. Physicians receive access rights to the hospital to see and treat patients.\textsuperscript{382} Their services are then charged separately to the government on a fee-for-service basis.

\textsuperscript{381} Appendix C section 1 discusses more fully these approaches and the literature critiquing each view.

\textsuperscript{382} The hospital as the physician’s ‘workshop’ best describes this relationship between physician and hospital. The workshop is kept available and tidy for physicians by the hospital. The physician then arrives on the scheduled days to carry out their tasks.
In Canada, physicians are reimbursed a fixed amount for services provided to patients. These fees are negotiated between the government and the medical association in each province. Each physician is granted a billing number within a certain city/region by the provincial government.

Hospitals, especially research and tertiary hospitals, also receive funds from other sources such as private donations and drug companies. The drug companies are especially generous in donating money to support research on the efficacy and effectiveness of their medications.

The ideas and approaches to health care delivery are important in framing the setting of technology. Issues and ideas such as health care spending, funding, and preventive versus acute delivery of health care are important to understanding the environment surrounding software development in both case studies.

11.5. Health Care Funding Debate

From the criticism and concern over health care spending has emerged several proposed funding and delivery approaches to health care to solve this apparent crisis. These old and new funding and delivery approaches include cost budgeting, free market, health valuation methods, and preventive medicine. The following section discusses these various approaches, and draws upon the literature to understand and evaluate each perspective.

\[\text{Recently, provinces has been restricting billing numbers in an attempt to contain health costs. In addition, provinces have restricted billing numbers within the cities in order to encourage physicians to move into the country where physicians are scarce.}\]
One method of allocating resources is to focus on money. There are various types of budgeting and monetary allocation; each type has its own assumptions about the hospital/community setting, each ignoring in a number of cases the realities of the health care setting (Evans 1984).

Most health regions/hospitals receive funding in a lump sum every year through "global budgeting". This form of payment is considered to ignore the treatment that occurs in the hospital during the year, causes a constant spiral upward on next year’s budget, and encourages the spending of surplus resources to maintain budget levels (Evans, 1984).

In Canada, physicians are reimbursed a fixed amount for services provided to patients. These fees are negotiated between the government and the medical association in each province. The concept of fee-for-service was to allow physicians to determine the appropriate use of services; a decentralized form of resource allocation. This has sometimes been called the naive-medico technical model of resource allocation (Evans 1984).

What has surprised many is that an increase in physician supply leads to more services provided, and the constant income of all physicians. This supplier-induced demand is a common pattern, and makes sense when you consider the inability of consumers to make many of the decisions concerning their health care. Combined with dramatic service utilization differences between regions and government deficits, this has prompted concern about the physicians' ability to determine needs, and to a number of alternative approaches to funding health care.

Other more “incentive reimbursement” systems (Evans 1984, p 174) have been
proposed, including Case Mix Groupings, Diagnostic Related Groups, or "episodic care". Case Mix Groupings provide funding directly or incrementally to hospitals based on their case-mix of patients during a year. Average lengths of stay and resources consumed are gathered from the hospitals, and then used to enforce those standards. Hospitals able to handle the case at a lower cost will produce "profits" and in a sense force out other hospitals. Implicit and weak assumptions of CMGs include:

(1) A hospital is a single unified entity interested in profit maximization, and having control over production. The loosely coupled and externally driven usage of the hospital by physicians is ignored (Evans 1984).

(2) Case Mix Groupings (CMGs) have few justified variations in resource use. Research suggests that this is not the case, and it is difficult to say that the less resource intensive patient is being treated more efficiently (Plomann & Shaffer 1984, pp. 38-48; Vladeck & Kramer 1988, pp. 339-340).

(3) CMGs have no direct measures of quality.

(4) CMGs ignore many unique patient characteristics in order to create a manageable number of categories and to allow comparison between patients.

It has been proposed that a free market system would solve the funding crisis, and allow consumers to choose their products and services and pay "out-of-pocket". This approach has been heavily criticized by health economists for the blind usage of traditional economic theory and inability to see the poor results of the closest free market health care system in the world: the United States (Rachlis & Kushner, 1994, pp. 152-161). Theoretical
criticism of traditional economic application in health include supplier induced demand, uncertainty of illness, and the externalities of illness (Evans 1984).

“The experience of the United States indicates that mixed systems of private and public funding support a continuing expansion of the health care sector relative to the rest of the economy ... First, more money spent on health care must mean less spent on something else...Second, the American experience makes it clear that private funding, at least through the private insurance mechanism, is far less efficient than public funding. A much higher, and rapidly growing, proportion of spending goes to pay for the administrative costs of the funding system .... Third, as private funding sources become a significant proportion of health expenditure, people with lower incomes and at higher risk of health expenditure (the two are correlated) are excluded from care ... Finally, a nominally private system still requires a large public component because the people with the greatest health problems—the elderly, the poor, those with certain chronic illnesses—must be covered by public programs...Private funding is a popular 'quack' remedy with dangerous ingredients which will defeat, not support, the objective of controlling costs” (Seaton et al., 1991, p B-82 83).

Evans (1984) provides an interesting insight into the inability of health care services to expand continuously when provided free:

“Indeed, the direct effects of most, if not all, health services on their users' well-being, independent of their anticipated health effects, is negative. Dentistry, drugs, diagnostic and therapeutic interventions, hospital stays, are frequently uncomfortable or frightening in and of themselves. Few, indeed, would choose to purchase them in the absence of an expected health benefit. And those who do—drug addicts, say, or sufferers from Munchausen’s Syndrome, who derives pleasure from medical care and counterfeit illness to get it are generally regarded as unwell. Anyone who seeks care when he is not sick, is sick” (p 5).

In fact, it has been those borderline free-market areas that have continued to increase income and expand their total take of the GNP, including dentistry (Evans 1984, p 87) and

The alternative to many of these methods is to start focusing on the final outcome: health and quality of life (Akehurst 1985; Williams 1985; Karam, Sundre & Smith 1986; Green & Kreuter 1991). Implicit in the approach is a need to determine a societal cost/benefit of intervening to improve the condition of a population. However, the complex environment and the effects of health programs and interventions on many populations raises the question, “whose benefits/costs”? (Evans 1984).

“Hence programs which seem obviously efficient from a society-wide perspective are frequently blocked because they impose losses on a powerful subgroup whose interests cannot or will not be compensated, while other programs whose costs viewed globally seem clearly beyond their benefits are adopted because of their partial payoffs. The cost-benefit analyst may treat all winners and losers equally; the political system does not” (Evans 1984, p 250).

The process of valuing life is also complicated by the insistence on human life as priceless, despite the limited resources available not only for health, but many other private and public goods. As a result, we assign infinite value and zero value to areas where the outcomes are essentially the same on health, largely because the fear of death from certain causes seems to overtake the fear of death from other causes (e.g. death in a plane versus a car) (Slovic, Fischhoff, & Lichtenstein 1980, 1982; Evans 1984; Slovic 1986, 1987; Slovic, Fischhoff, & Lichtenstein 1988; Slovic, Kraus & Covello 1990).

---

384 Rachlis & Kushner (1994) give a stinging criticism of the unregulated pharmaceutical industry in a chapter titled “Unhealth Alliance: Medicine and the Drug Industry”. They quote a number of physicians who suggest that between one and two-thirds of antibiotics are unnecessary or inappropriately used, and that the thousands of elderly die each year of drug side effects. Most of the problem appears to be physician lack of knowledge about drug products, the marketing misinformation of drug companies, and the uneasy union between drug companies and academe.
Life-year measures, number of years added by an intervention, and quality of life years add a subjective assessment of quality to those increased years. This approach has also been criticized for its subjectivity as illustrated by the wide disparity in life valuation between healthy observers, who value the patient's condition lower than patients of the disease (Mulkay, Ashmore, & Pinch, 1987). Health valuation has also been criticized for its cost (Bostrom & McDonald 1995). Nevertheless, the health valuation and the analysis of outcome is a step in the right direction as groups examine the patient outcomes of care, and address the wide variations in practice patterns (Rachlis & Kushner 1994, pp. 91-123).385

The goal of health care is to prevent disease-related death (mortality), and disability (morbidity). As mentioned earlier, there are two approaches to reducing mortality and morbidity: preventive and curative medicine. Preventive medicine (PM) attempts to prevent disease in patients before the disease occurs, or to detect disease early enough to prevent much of its potential harm. Curative medicine (CM) deals with disease in the final stages when the symptoms and effects of the disease appear. Treatment involves surgery, drugs, or both.

According to several researchers, preventive medicine is over-taking curative medicine in importance for a number of reasons:

1. The relative number of diseases that can benefit from curative medicine has decreased:

"The role of behavior in health has received increasing attention since the turn of the century, as the causes of death in the United States shifted from infectious and dietary disease to chronic diseases. Mortality from most of

385 Various terms are being used in hospitals such as TQM (total quality management) and CQI (continuous quality improvement) to reflect this change in perspective.
today's ten leading causes of death...could be substantially reduced if people would improve a few health related behaviors” (Sarafino 1990, p. 176)

(2) An ounce of prevention is worth a pound of cure and therefore preventive medicine will decrease health costs.\textsuperscript{386}

(3) Some diseases strike quickly, and patients die before curative medicine can intervene:

“...the majority of persons who die of a heart attack do so before they reach medical care” (Vogt 1983, p144).

(4) Health status of the population is expected to increase under preventive care by avoiding the danger of acute illness.

Currently, with 73\% of resources being consumed by hospitals and physicians, there is a power struggle occurring in most provinces over the redistribution of resources into preventive areas. In the province of British Columbia, regional funding versus hospital funding, and hospital closures have attempted to move funding away from these acute care institutions, and have resulted in anger and strife.\textsuperscript{387} The result has been a somewhat chaotic change in focus by governments toward preventive medicine (Green 1996).

\textsuperscript{386} This view has been challenged by a number of authors who argue about the costs of prevention (Russell 1986).

\textsuperscript{387} A recent button I found in the cafeteria of the hospital said “Some Cuts Don’t Heal”. Recent advertisements by the physicians’ medical association warned of the dire consequences of cuts to medical care and their ability to provide that care.
12. Appendix D: Contextual Information for Case Two - EMPOWER

12.1. Introduction

Building software to address the planning of mammography screening required the study of two research areas. The first was breast cancer epidemiology and mammography screening, and the second was a health planning model called PRECEDE/PROCEED.

12.2. Breast Cancer Epidemiology and Breast Cancer Screening

Overview: There are two research areas and questions involved in breast cancer: the factors affecting the disease itself, and factors affecting breast cancer screening for the early detection and treatment of the disease. While the focus of heart disease was on the behavior and environment affecting heart disease, and how to change those behaviors, very few modifiable behaviors and environmental factors have been found to affect breast cancer. Hence, the focus has shifted to secondary prevention through early mammography screening, detection, and treatment of the disease.

Breast cancer, like any cancer, occurs when cells grow at abnormal rates. This is caused by mutations and damage to the cells’ genes that regulate cell division (Davis & Bradlow 1995). The spreading of these cells into the rest of the body causes death. In the United States, 180,000 new cases and 46,000 deaths occur annually (Doyle 1995).
Like cardiovascular disease, most risk factors of breast cancer are highly inter-related and causality is difficult to prove. No one factor seems to be the starting point for the other risk factors, the cause of intermediate effects on the breast tissue, or the final cause of breast cancer.

Unlike cardiovascular disease, most of these risk factors are unmodifiable. These risk factors include age, early onset of menstruation, late entry into menopause, having children late in life, family, never having had or breast fed a child, or family history of disease (School of Public Health 1994; Davis & Bradlow 1995).

So far, research has only managed to disprove and weaken a number of proposed risk factors. High fat diet and smoking have been ruled out as causes, and the debate over environmental pollution and pesticides continues (School of Public Health 1994; Davis & Bradlow 1995). Even family history, the usual risk factor in many diseases, turns out to explain very few of the breast cancer cases (School of Public Health 1994).

A lifestyle factor, associated with 50% increases in risk of BC, includes alcohol consumption.

One clinical factor linking many of these risk factors is the level of a woman's exposure to the hormone estrogen.\textsuperscript{388}

"The hormone influences cell growth by binding to an intracellular protein known as the estrogen receptor. Complexes of hormone and receptor can bind to DNA in the nucleus and activate genes that direct cell division. Such activation speeds the rate of DNA replication and so increases the likelihood that a mutation, possible carcinogenic, will arise and go unrepaired" (Davis & Bradlow 1995).

\textsuperscript{388} The estrogen and alcohol are viewed differently in heart disease, both acting as a defense against heart disease. Physicians in the two heart clinics prescribe estrogen supplements to female patients to treat heart disease.
The estrogen connection would explain many of the risk factors. However, the result of the risk factor studies suggests that there is little that can be done to prevent breast cancer. Therefore, the only solution is to detect and treat the disease as early as possible.

Detecting breast cancer through a **screening mammography**, a procedure that scans the breast for abnormal growth, appears to be one method of reducing death from breast cancer. If a growth is detected, a more thorough **diagnostic mammogram** is performed to evaluate the cancer fully. If a growth is detected, a more thorough **diagnostic mammogram** is performed to evaluate the cancer fully.

The only effective treatment of breast cancer is a removal of part or all of the breast through a **lumpectomy** (removal of small lump) or **mastectomy** (removal of the breast). With a **radical mastectomy**, the breast, the lymph nodes in the nearest armpit, and the muscles linking the upper part of the chest with the shoulder are removed. The purpose is to ensure that any cells that have spread to these regions are removed. A **modified radical mastectomy** leaves the muscles intact.

The regular participation in mammography screening is expected to reduce the breast cancer mortality in women ages 40 to 49 by 24%, and 31% for women 50 years and older (Salazar & de Moor 1995). These benefits, however, are not appearing as women continue to avoid mammography screening. Only 38% of women over the age of 40 have ever had a mammogram in the United States, and only 15% have had a mammogram in the previous year (Salazar & de Moor 1995). This suggests that 85% of women are currently falling

---

389 Other methods such as **breast self examination** by the women, and **clinical breast examination** by the physician have been criticized as ineffective at spotting abnormal tissue in early stages, and are discouraging women from seeking **mammography screening** (McCool 1994)
outside the regular mammography examination required to effectively detect and treat breast cancer.

Factors affecting breast cancer screening include primary (women), secondary (physician), and environmental (availability of mammography equipment) targets. Since mammography screening is a medical intervention, the physician consulting the patient is important in mentioning and directing patients into screening.

**Difficulty of Arranging treatment, fear of cancer/treatment, cost, accessibility and convenience, time, other means of knowing, and health provider suggestion** have been researched as factors affecting the decision to seek mammography screening (McCool 1994; Salazar & de Moor 1995). Older women are especially absent from mammography centers, believing that cancer is not a health risk, and that mammography was only necessary if symptoms were present (McCool 1994). The fact that over 50% of breast cancer deaths are found in women over the age of 65 suggest a cause and/or need to address this population.

The causes of breast cancer are largely unmodifiable, creating a need for secondary prevention (the early detection and treatment), versus the primary prevention of disease. Detection and early treatment through lumpectomy and mastectomy are the only known method of preventing death from the disease. However, participation in mammography screening is low, especially amongst the older women most at risk of the disease. Factors affecting the choice to have a mammogram include difficulty of arranging treatment, fear of cancer/treatment, cost, accessibility and convenience, time, other means of knowing, and health provider suggestion.
12.3. Health Planning

There is a need to understand how to plan health promotion programs in order to improve health and quality of life. This became especially important in the EMPOWER case study in modifying and developing a planning tool. The following discusses the difficulty of planning health programs, the health promotion model employed in this study (PRECEDE/PROCEED), factors in the model, the stage of planning, points of planning failure addressed by the model, and the meaning of the linkages between factors.

Planning interventions to improve health is difficult, and fraught with common and unique pitfalls as compared with other planning activities (Nadler 1982). It became important to have a good method of analyzing and planning the software development process with some health promotion model, to ensure that the software product would eventually address health related issues in a relevant manner.

As a result, both case study one (SoftHeart) and especially case study two, have been influenced by the PRECEDE/PROCEED model of health promotion planning (Green 1974; Green & Kreuter 1991). The model has been used in over 500 published applications, and is perhaps the most widely used model of health planning. The model is based on theory, research, practice, policy, and case studies and evaluation within health promotion. Personal experience working with and teaching the model has provided me with an intimate knowledge of its workings.

PRECEDE stands for “predisposing, reinforcing, and enabling constructs in educational and environmental diagnosis and evaluation”. PROCEED stands for “policy,
regulatory, and organizational constructs in educational and environmental development”. As the verbs suggest, PRECEDE is the necessary stage of thinking, analyzing, and weighing alternative explanations and programs before PROCEED-ing to carry out the intervention. PROCEED includes both implementation and evaluation phases. The following provides a graphical depiction of the causal linkages and stages of PRECEDE/PROCEED:

Figure D-1: PRECEDE/PROCEED Model


The model attempts to simplify the complex domain of health and health promotion planning by using various factor categories and linking these categories in a causal chain. These factors include quality of life, health, environment, behavior, predisposing, reinforcing
Appendix D: Contextual Information in the EMPOWER Case

and enabling factors, health education, and policy regulation and organization.

The purpose in each stage is to evaluate the characteristics and dimensions of each factor category (quality of life, health, etc.) relevant to a specific target population. The purpose is to employ a population-based versus a resource-based approach to planning to ensure the health program is driven by the needs of the population. Therefore, identification of the target population is the first step in building the plan. Factors are then ranked according to importance and changeability in order to determine those areas of focus. Serving as a means-ends chain, this process yields those ends and important means to those ends that will be targeted by an intervention.

Social diagnosis identifies those issues of life quality most important to the target population. As a part of obtaining this information, the PRECEDE model encourages community and stakeholder participation in identifying these needs. Epidemiological diagnosis identifies those health issues that affect quality of life. Ranking, prioritizing, and setting goals for the health issues produces the target health issues(s) for the project. Behavioral and Environmental diagnosis then identify those individual behaviors and environmental factors that influence health directly or indirectly through its influence on behavior. Ranking, prioritizing, selecting, and setting goals for behavior and environmental factors is then completed. Educational and Organizational diagnosis then identifies those predisposing (attitudes, beliefs, knowledge, perceptions), reinforcing (attitudes and behaviors of peers, family, health professionals, etc.), and enabling (availability of resources, accessibility, referrals, rules or laws, skills) factors that affect behavior and environment.
Ranking, prioritizing, selecting, ordering in a causal chain, and setting goals for the predisposing, enabling, and reinforcing factors is then completed. Finally, *Administrative and Policy Diagnosis* identify current and desired health education, policies, regulations, and organization that support current and future levels of predisposing, reinforcing, and enabling factors.

One of the difficult aspects of health promotion planning is evaluating the outcome. This is especially true for larger interventions that have longer-term implications for health that cannot be measured immediately. Therefore, the model suggests several types of evaluation, including *formative evaluation* (program development), *process evaluation* (implementation), *impact evaluation* (predisposing, enabling, reinforcing, behavior and/or environment), and *outcome evaluation* (health and quality of life). Each step of evaluation is an important component of the PROCEED steps to carrying out and monitoring the effect of the intervention.

The model and workshop discussions about the model (Weiss 1974) also indicate where a project may fail. The first is *program failure* where the program is poorly conceived, developed and implemented. The second is *theoretical failure* where the program was appropriately constructed, but the theory is poor. The third and final problem is *measurement failure* where the program and theory were correct, but the measuring instruments were unable to detect the problems.

In evaluation, it is also important to determine several important characteristics of the intervention (Green & Kreuter 1991). These include the *effective dose* as the threshold level
required to achieve the effect, the *lethal dose* as the point of diminishing returns before the intervention would begin to cause damage, and the *therapeutic index* which is the difference between the effective dose and the lethal dose. All three concepts suggest a need for evaluation to search out unexpected negative consequences of the program.

One of the more interesting aspects of the model is the linkages between the factor categories, and the important meaning behind each linkage. First, both behavior and environment are tied directly to quality of life, and indirectly to quality of life through health. This is perhaps the key issue in health promotion planning and health care delivery; that changes in behavior and environment cause immediate changes in quality of life, while also affecting quality of life through usually later changes in health.\(^{390}\) Two, behavior influences environment, because individuals and groups can change their environment and/or influence it to some extent. Three, enabling factors are the only factor tied to environment, suggesting that the resources, skills, and organizational factors are crucial to changing the environment. Fourth, enabling factors can influence reinforcing, and reinforcing influences predisposing factors. However, the reverse does not directly hold. In the longer term, predisposing factors may influence behavior, which in turn can be reinforced. This suggests that predisposing factors need to be dealt with and changed where necessary before any predisposing and reinforcing change is possible. This point is made clear in the various "triage" models used throughout the book in clinical, worksite, and community settings, where predisposing factors are addressed first before any reinforcing and enabling factors are considered. Fifth, health

\(^{390}\) Smoking is an example of a behavior that when reduced, is usually accompanied by severe withdrawal in the short term, and improved health and quality of life in the long term.
education can influence policy, regulation, organization, predisposing, reinforcing, and enabling factors, and is therefore the primary method of intervening. Sixth, policy, regulation, and organization influences the framework indirectly through health education (predisposing, enabling, and reinforcing factors) and directly through enabling factors.

Understanding the causal chain of program intervention to health and quality of life has been important in both case studies. Planning health promotion interventions requires a good planning framework. The PRECEDE/PROCEED model of planning has influenced design in both case one and case two through its language, theory and organized method of planning.
13. Appendix E: Pictures of SoftHeart

The following screens are from the SoftHeart program. The interface consists of 18 possible display panels, 14 input screens, and numerous dialogue boxes for information searching and displaying. Those panels available for viewing and editing are determined by the user's login identification.

The introductory screen asks for user login and identification.\(^{391}\)

Figure E-1: Introductory Screen

\(^{391}\) The dark boxes protect the identity of the organization.
Immediately afterward, a patient search box asks for the first patient to view. Users can type names and/or use the ‘browse’ window to search and double click on a selection.

**Figure E-2: Patient Search Window**

The main screen opens immediately following the selection of the patient. The main screen shows up to 18 possible panels, starting with patient information, with “tab” buttons along the side for viewing different information about a patient. Tabs with and asterisk (*) allow the user to edit the information in the panel. Tabs with trailers (…) when pressed, take the user to a special dialogue box for more detailed viewing and data
input. The “screen”, “face”, and exit icons at the bottom are push buttons that allow the editing of the panel’s information (depending if there is an asterisk beside the tab button), selection of a new patient, or exiting. The following shows the main screen when first entering the software, and shows patient information:

Figure E-3: Patient Information Panel

A panel displays either single or continuous information about the patient. The patient information panel displays information about the patient, including special ‘browse’ windows for viewing and scrolling through information about the patient’s schedule and their associated physicians. Clicking and highlighting different physicians in the window
Appendix E: Pictures of SoftHeart

causes the information to the left to be updated with that physicians address, phone number, etc.

Clicking on a tab button on the right displays a different panel, in this case lipid information.

**Figure E-4: Lipids Information Panel**

<table>
<thead>
<tr>
<th>Yr Dt Risk%</th>
<th>7.8 Aug 21, 95 6.4</th>
<th>4.4 Apr 29, 92 13.5</th>
<th>2.7 Aug 9, 90 16.9</th>
<th>2.0 Oct 26, 89 10.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Chol:</strong></td>
<td>&lt; 6.2 6.24 5.00</td>
<td>7.80 6.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HDL Chol:</strong></td>
<td>&gt; 0.9 0.95 1.00 1.18 1.10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Trig:</strong></td>
<td>&lt; 2.3 3.80 3.03</td>
<td>3.21 2.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LDL Chol:</strong></td>
<td>&lt; 3.4 4.38 4.30</td>
<td>4.99 4.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>APOB:</strong></td>
<td>0.67-1.17 1.30 1.45</td>
<td>1.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>APOA:</strong></td>
<td>1.10-1.30 1.40 1.69</td>
<td>3.44 3.44</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LDL-C/APOB:</strong></td>
<td>1.32 2.69 3.44</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LPa:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Epheno:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Partial***

**Lab Results***

**Ready: Push Button for Action**

**Event:** Records: 20662/20662 **Record Unlocked** **Time:** 7:22:44 pm
All continuous information is displayed in a spreadsheet look-and-feel with the most recent information displayed first. Clicking on the arrow keys takes you back to previous values. Another example of continuous information is the medications panel:

**Figure E-5: Medications Information Panel**

<table>
<thead>
<tr>
<th>MEDICATION</th>
<th>MICHELLE CHIASSON</th>
<th>Yr Dt Flak:</th>
<th>7.8 Aug 21,95 6.4</th>
<th>4.4 Apr 2,92 13.5</th>
<th>2.7 Aug 9,90 16.9</th>
<th>2.0 Oct 26,89 10.5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Dose</td>
<td>Freq</td>
<td>Dose</td>
<td>Freq</td>
<td>Dose</td>
</tr>
<tr>
<td>Questran</td>
<td></td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Gemfibrozi</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>600</td>
</tr>
<tr>
<td>Niacin</td>
<td></td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lovastatin</td>
<td></td>
<td>-</td>
<td>-</td>
<td>30</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RESTORIL-T</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
</tbody>
</table>

This panel displays medications prescribed and currently taken during a patient visit. For example, on October 26, 1989, the drug Questran was taken by the patient in 4 grams amounts three times daily, which was then lowered to twice daily. The next visit on August 9, 1990, showed the patient was taking the drug twice daily. But lab results may have prompted the physician to raise the dosage once again to three times daily.
Clicking on blue 'screen' icon at the bottom for panels that contain an asterisk (*) allows users to edit information. The following is the edit screen for the lipids panel:

**Figure E-6: Lipids Edit Panel**

<table>
<thead>
<tr>
<th>Lipids</th>
<th>Controls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yr Dt Risk:</td>
<td>7.8 Aug 23</td>
</tr>
<tr>
<td>Total Chol:</td>
<td>LDL Chol:</td>
</tr>
<tr>
<td>HDL Chol:</td>
<td>LDL/HDL</td>
</tr>
<tr>
<td>Trigs:</td>
<td>ApoB:</td>
</tr>
<tr>
<td>LDL Chol:</td>
<td>ApoA:</td>
</tr>
<tr>
<td>Lp(a):</td>
<td>Epheno:</td>
</tr>
</tbody>
</table>

**LAB RESULTS**

- Add event
- Del event
- Exit

**DATA INPUT**

- Lab: Record 20552/20561
- Record Locked
- Data Entry
- Data Entry
- Exit

**Microsoft FoxPro**

**Pat info**

- Patient Add
- Lab Results
- Diet/Ex/Wght
- Smoke/Alch/BP
- Stress Test
- Meds/Side
- Scheduling
- Physicians
- Reports
- Trends
- Databases

**Time:** 7:27:01 pm
The following is the edit panel for the patient information panel:

**Figure E-7: Patient Information Edit Panel**
There are also special information panels such as the risk calculator and the trends panel for graphing. The following shows the trends panel:

Figure E-8: Trends Information Panel

Different values to be graphed are selected from the list box above.
Finally, there are special panels marked with "..." that take the user to a series of special screens for input and display. The following is the Scheduling... panel:

Figure E-9: Scheduling Panel
The following is the Reports... dialogue boxes for access and printing of any reports:

**Figure E-10: Reporting Panel**
14. Appendix F: Pictures of EMPOWER

The following screen pictures are from the EMPOWER program. The program consists of six boxes, with approximately 80 pages of information to view, and data to enter by the user about his/her mammography project. The following is the introductory screen asking for user identification:

**Figure F-1: Introductory Screen**
Once the user has identified and picked or typed in a new project name, EMPOWER’s assumptions screen is displayed:

**Figure F-2: Assumptions Screen**

- The user is interested in developing a program to meet the unique needs of a particular population which we will refer to as a community.
- The user is most likely to be a member of a small group with an interest in developing or improving a breast cancer screening program. This group constitutes a core planning group.
- The planning and implementation steps of EMPOWER will lead to programs and outcomes that can be evaluated.
- When the term "we" is used, it refers to your small planning group and the experts represented by EMPOWER.
- Users of this application of EMPOWER have a charge, mandate, responsibility, or opportunity to plan and implement a program of early detection of breast cancer.
- Planning and implementation of actions will require the participation of agencies and organizations representing multiple sectors within the target community.
- This application of EMPOWER is intended for local rather than national, state, or provincial level planning.
After reading a few additional screens of information, the user is presented with the planning model, which also serves as the global map page to the six modules:

Figure F-3: Global Map Page
Users can select any of the six modules by clicking on the boxes (situational analysis is selected by clicking the background). The local map page of the situational analysis is displayed after clicking the background:

**Figure F-4: Local Map Page of the Situation Analysis**
Each box is then an entry point to a page of information and/or data input. For example, clicking on the “shared vision” produces the following screen:

**Figure F-5: Shared Vision Statement**

A statement of your vision for this project is needed to keep you and others working with you focused on what you mutually want to achieve for this community. This initial vision statement may be revised and updated while you are working on your project.

As you incorporate feedback from others, this will become a "shared vision."

**A community free of needless suffering due to breast cancer.**

Users can then click on the bottom blue window to produce a dialogue box for changing the vision statement.
Clicking on the “role in the project” box produces the following screen:

Figure F-6: Role in the Project

From the options below, choose the one that best describes your role in this application of EMPOWER.

- An official representative of an agency or organization
- A community member not representing an organization
- A trainee, learner, or student
- An outside consultant
Users select an item that best fits their project. The following screen shows the “program goal” page:

Figure F-7: The Program Goal Page and Consult-on-Tap Resources

A statement of your program goal for this project is needed to keep you and others working with you focused on what you mutually want to achieve. This is a default program goal that you should edit to be consistent with the population you defined on the previous screen. This initial program goal may be revised and updated as you progress with your planning.

The morbidity and mortality associated with breast cancer will be reduced through increased screening.

[ Click on this box to revise your program goal ]
This page also shows the "consult-on-tap" resources menu. This menu takes a user to various information pages and dialogue boxes. The first section of the menu are resources for the page. The following "why" help is for the program goal page:

Figure F-8: "Why" Help

Program goals expand your vision statement by indicating the "who" and "what". By refining this goal, you and others working on this planning effort will have a clearer idea of the purpose of the program and its direction. Like the vision statement, program goal cancer will be reduced through increased screening.

[ Click on this box to revise your program goal ]
The next picture shows the “other references” dialogue box for this page:

Figure F-9: Other Reference Help

Click on this box to revise your program goal.
The bottom section of the "consult on tap" menu includes general information. The following picture shows the "data files" available:

**Figure F-10: Data Files**

<table>
<thead>
<tr>
<th>File Navigation</th>
<th>Consult on Tap</th>
<th>Tutorials</th>
<th>Local Data</th>
<th>Options</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responding Here</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>projectNotes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>References</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case Example</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Glossary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Files</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- University Research Centers
- Non-University Research Centers
- PRECEDE/PROCEED References
- Community and Governmental Organizations
- Clinical Oncology Center
- Combined Health Information Database
- Intervention Strategies
- Year 2000 Objectives

The morbidity and mortality associated with breast cancer will be reduced through increased screening.

[ Click on this box to revise your program goal ]
Selecting a data file opens Windows help files, or special books that display and allow the editing of information. The following is the university research centers book:

**Figure F-11: University Data File**

<table>
<thead>
<tr>
<th>Name</th>
<th>McDowell Cancer Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>800 Rose St</td>
</tr>
<tr>
<td>City/State/Zip</td>
<td>Lexington, Ky 40536</td>
</tr>
<tr>
<td>Phone</td>
<td>(606) 233-5541</td>
</tr>
</tbody>
</table>
Back to the situation analysis local map, the final box shows the redesigned spreadsheet “look-and-feel” for a stakeholder assessment of “current resource and planning capacity”:

**Figure F-12: Current Capacities and Resources**

Rate on the scale of High, Medium or Low each of the groups within this category that you wish to include in your planning effort.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumer groups</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self care groups</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Health care professionals</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Social service professionals</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other professionals</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Private business</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local health departments</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other public agencies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mass media</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other stakeholders</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local hospitals and health centres</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voluntary agencies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A summary report can then be displayed and printed:

**Figure F-13: Summary Report for Situation Analysis**

![Summary Report for Situation Analysis](image)

Data from this module is then displayed and accessible from the other five modules.
15. Appendix G: The SoftHeart Program

The following appendix describes the structure of the SoftHeart program, and provides a glimpse of the code.

SoftHeart was developed in Foxpro 2.6 for Windows. The program currently is around 32,000 lines of code. Foxpro is structured in the traditional manner, with a separation of data (see Figure 4.1 for SoftHeart's data structure) and program. There are four types of programs: header, screen, and reports and labels. Header and screen programs use the same text editing files, but most of the screen code is generated by a screen designer. However, to do many of the display features including imbedded browse windows and screen refreshing, I went directly into the screen code and modified. The following figure shows the structure of header and screen programs:
The header-screen structure is repeated throughout SoftHeart, including the large View Program and Display Screen Program that produces and runs all the information panels. The Display Screen Program alone has 3,640 lines of code.

SoftHeart is started by a Setup Program that sets the operating environment, and opens the login database files for the login screen. The Setup then calls the View Program to obtain user login information, attach a code library, load the data files, and then call the Display Screen Program to open the information panels. The following is the code for the beginning of the View Program, up to the start of the push button function (dots ... are placed where the code was shortened):

**DSVIEW.PRG**

**Purpose: Physician viewing of patient information.**
** Called by: Icon
** Calls to:
   * SPR
   * PRG
   * aadata.prg to open databases

** Created: May 30, 1995

* Open Introduction Screen
  DO dsintro.spr

* Set the global string for push buttons
  k_contstrl = ""
  k_role = ""

* SECURITY CHECK and pick up button options
  DO inuser.prg

* Set up array for windows
  DIMENSION k_wlist(1)

* Create a list of edit screens
  l_x = 0
  IF AT("INFO", UPPER(k_contstr1))<>0
   l_x = l_x + 1
   DIMENSION k_elist(l_x)
   k_elist(l_x) = "PATIENT INFO"
  ENDIF

* Open the Log file (k_log set in aasetup by DEFAULT database file).
*IF k_log
   IF NOT USED("log")
    USE log IN SELECT(SELECT(1)) AGAIN SHARE
   ENDIF
   SELECT log
   APPEND BLANK
   REPLACE log.uusernumb WITH ctlusers.uusernumb
   REPLACE log.ufhame WITH ctlusers.ufhame
   REPLACE log ula name WITH DATE()
   REPLACE log.lgstime WITH DATE()
  ENDIF

* Attach general purpose functions and controls
  SET PROCEDURE TO aagen.prg

* Open databases and establish linkages
  DO aadata.prg

* Set filter depending on role (Lipid, Healthy Heart, Both)
* Note: k_role set in the insecur.prg, will also be used in scheduling
IF ALLTRIM(UPPER(k_role)) == "L"
    SELECT patient
    SET FILTER TO plipidec = .T.
ENDIF

IF ALLTRIM(UPPER(k_role)) == "H"
    SELECT patient
    SET FILTER TO phheart = .T.
ENDIF

* Set all p_ variables to private

PRIVATE ALL LIKE p_*

* Create global variables

    * Global Variables

    k_numberscreens = 17

* Set up the on error routine

    ON ERROR DO p_error WITH ERRORQ, MESSAGE()

* Attach the graphing library

    SET LIBRARY TO gswfil.fll

* Initialize Screen Refresher Array

    FOR l_i = 1 TO k_numberscreens
        STORE .F. TO k_screen (l_i,1)
        STORE 0 TO k_screen (l_i,2)
    NEXT

* Search for the first patient

    DO hhpatsrh.prg
    k_displaywind = "w_patchart"

* Set UP patient global variables

    DO p_patvars WITH patient.pkey

* Disable and establish certain keys

    PUSH KEY CLEAR
    ON KEY LABEL ESC DO p_nothing
    ON KEY LABEL MOUSE DO p_window WITH WONTOP()
    ON KEY LABEL SHIFT+F4 DO p_file
    ON KEY LABEL SHIFT+F5 DO p_oneontop
    ON KEY LABEL CTRL+F4 DO p_closewind
    ON KEY LABEL CTRL+F5 DO p_getwind

* Open SPR file

    DO dsview.spr

* Release variables and clear

    DO
CLEAR ALL
CLEAR WINDOWS
CLOSE ALL
CLEAR

ON ERROR
POP KEY ALL
* Return
RETURN .T.

* PUSH BUTTON FUNCTIONS AND PROCEDURES for DSVIEW.SPR

FUNCTION fvcontrol
PARAMETERS pcontrol, p_value && ** ADDED p_value for graph1 and graph2
l_advance = 1.693

k_pkeyold = patient.pkey

DO p_gtusage WITH p_control

DO CASE

CASE AT("-", UPPER(p_control)) <> 0
    RETURN .T.

CASE AT("LEFT.BMP", UPPER(p_control)) <> 0
    IF k_uppervisit = patient.penumb
        =f_message(22)
        RETURN .T.
    ENDIF
    IF k_uppervisit+4 <= patient.penumb
        k_uppervisit = k_uppervisit + 4
    ELSE
        k_uppervisit = patient.penumb
    ENDIF
*  MOVE WINDOW (k_displaywind) TO -0.8,-1.0
    SHOW GETS WINDOW (k_displaywind)
    RETURN .T.

CASE AT("RIGHT.BMP", UPPER(p_control)) <> 0
    IF k_uppervisit <= 4
        =f_message(24)
        RETURN .T.
    ELSE
        k_uppervisit = k_uppervisit - 4
    ENDIF
*  MOVE WINDOW (k_displaywind) TO -0.8,-1.0
    SHOW GETS WINDOW (k_displaywind)
    RETURN .T.

....
Data input modules are handled separately with special input and screen programs. The programs contain functions attached to data input fields and push buttons. The code is activated when the buttons are pushed. The event code is stored entirely in the Input Program to avoid mixing pure screen and draw information with action-oriented code. This was based on the suggestion of Goley (1993). The following is the input program and the display screen for the lipids panel:

**INLIPIDS.PRG**

```plaintext
*** INLIPIDS.PRG ***
*** Created June 19th 1995 ***
*** By <name>***
*** Called by aaview.prg
*** Used to edit and save information on Lipids Panel.

*Local control variables
kl_cont1 = ""
kl_cont2 = ""

*Determines if changes have been made to a single record
kl_change = .F.

SELECT event
SEEK STR(patient.pkey,7,0)
SKIP k_upppervisor - 1

*Select lab and scatter memory variables to be edited (allows cancelling)
SELECT lab
SCATTER MEMVAR MEMO

IF EOF(Lab)
    k_recexists = .F.
ELSE
    k_recexists = .T.
    IF NOT f_locking() && Call to general f_locking function
        l_response=f_message(53) && OK dialog box
        * =f_message(52, "Exiting Edit") && Unable to lock most recent event. Returning...
        IF l_response = 1
            RETRY
        ELSE
            kl_cont = .F. && 01/31/96 kl_cont added to exit loop.
            RETURN .T. && Return to the main viewing READ
            ENDF
    ELSE
        kl_cont = .F. && 01/31/96 kl_cont added to exit loop.
    ENDIF
ENDIF

* Added to the following for the delete button
```
l_res = f_chkdata() && Determine if there is data for the event.

IF l_res
    && Set whether Del event will be activated
    kl_delstr = "\"
ELSE
    kl_delstr = "<"
ENDIF

** Add default movement for the kl_nextedit to Lab Results
* Movement to the next element on the list

l_elem = ASCAN(k_elist, "LIPIDS")
l_alen = ALEN(k_elist)

IF l_elem < l_alen
    kl_nextedit = k_elist(l_elem+1)
ENDIF

* Call the display screen for editing information.
DO inlipids.spr

RETURN .T.

FUNCTION: l_vcontrol
PARAMETERS: p_control: Button pressed
RETURNS: .T. once completed
CALLED BY: @GEt push buttons in inlipids.spr
Local control procedure for push buttons.

FUNCTION l_vcontrol
PARAMETERS p_control, p_recur

DO p_gtusage WITH "l_vc - INLPIDS " + p_control + ""

DO CASE

* Save Routine. Called by other push buttons.
CASE UPPER(p_control) = "SAVE"

IF NOT kl_change
    RETURN .T. && No changes made
ENDIF

l_cont = f_reccheck() && Checks the record for any omissions
SELECT lab
IF l_cont

    IF NOT k_receixsts && This record has not been created yet.
        IF NOT f_receixsts() && Check if added since beginning
            APPEND BLANK
            =f_locking()
            GATHER MEMVAR MEMO
            REPLACE lab pkey WITH patient pkey
            REPLACE lab numnum WITH event numnum
            REPLACE lab lcreated WITH DATE()
            k_receixsts = .T.
        ELSE
            =f_message(21)
    ELSE
        RETURN .T.
ENDIF

ELSE
    RETURN .T.
ENDIF

ENDCASE

RETURN .T.
Appendix G: The SoftHeart Program

SCATTER MEMVAR MEMO
RETURN .T.
ENDIF
ELSE
GATHER MEMVAR MEMO && Record exists, so store values.
ENDIF
ENDIF

REPLACE lab.lmodify WITH DATE() && Update modification date
UNLOCK && Unlock the record
SCATTER MEMVAR MEMO && Scatter new values.

k_screen (4,1) = .F. && Refresh display screen
kl_change = .F.
SHOW GETS && Redisplay calculated information on the edit panel.
=f_message(18)
RETURN .T.

* When the Exit push button is pressed...
CASE UPPER(p_control) == "EXIT"
 = l_vcontrol ("SAVE") && Save any changes if pending
* If the record exists, unlock it
UNLOCK
CLEAR READ

* Refresh the screen with any changes
SHOW GETS LEVEL 1 WINDOW w_lipids && skips event to k_uppervisit
kl_cont = .F.

CASE UPPER(p_control) == "NEXT"
IF AT("LIPIDS", UPPER(kl_nextedit)) <> 0
RETURN .T.
ENDIF
IF kl_change
 =l_vcontrol("SAVE")
ENDIF
CLEAR READ
RELEASE WINDOW w_edit
RELEASE WINDOW w_ineditcn
kl_nextedit = UPPER(p_recur)
RETURN .T.

* When the left push button is pressed...
CASE AT("LEFT.BMP", UPPER(p_control)) <> 0
IF k_uppervisit = patient.penumb && Already at upper visit
 =f_message(22)
RETURN .T.
ENDIF

** Save any changes to the record before moving on
 =l_vcontrol("SAVE")

k_uppervisit = k_uppervisit + 1 && Set upper visit
SELECT event
IF NOT EOF() & & Skip 1 since event database in order.
ENDIF
IF pkey = patient.pkey & & IF the next event record is still this patient...
SELECT lab & & Scatter lab information
IF NOT f_locking() & & Lock if a lab record exists.
l_response=f_message(53)
if l_response = 1
RETRY & & Try locking the file again if user wants
ELSE
SELECT event & & Asim Ranjha (1/31/96)
SKIP -1
k_uppervisit = k_uppervisit - 1
RETURN .T.
ENDIF
ENDIF
SCATTER MEMVAR MEMO

** Added a check to determine whether Delete Event button disabled
l_res=f_chkdata() & & Checks whether data for this event.
IF l_res
SHOW GET kl_cont2,2 DISABLE
ELSE
SHOW GET kl_cont2,2 ENABLE
ENDIF
SHOW GETS ' & & Refresh panel.
kl_change = .F.
ELSE
=f_message(23) & & Otherwise, display error message no more additional events.
SKIP -1
ENDIF
RETURN .T.

* When right push button is pressed, see left push button comments.
CASE AT("RIGHT.BMP", UPPER(p_control)) <> 0

IF k_uppervisit < 2
=f_message(24)
RETURN .T.
ENDIF

** Added SAVE routine to save any changes
=l_vcontrol("SAVE")

k_uppervisit = k_uppervisit - 1
SELECT event
IF NOT BOF() SKIP -1 & & Skip up 1.
ENDIF
IF pkey = patient.pkey & & IF the next event record is still this patient...
SELECT lab & & Scatter lab information
IF NOT f_locking() & & Lock if a lab record exists.
l_response=f_message(53)
if l_response = 1
RETRY & & Try locking the file again if user wants
ELSE
    SELECT event && Asim Ranjha (1/31/96)
    SKIP 1
    k_uppervisit = k_uppervisit + 1
    RETURN .T.
ENDIF

SCATTER MEMVAR MEMO

** Added check for event data to determine if Delete event button enable.

l_res=f_chkdata() && Checks whether data for this event

IF l_res
    SHOW GET kl_cont2,2 DISABLE
ELSE
    SHOW GET kl_cont2,2 ENABLE
ENDIF

SHOW GETS && Refresh panel.

kl_change = .F.

ELSE
    =f_message(24) && Otherwise, display error message most recent event.
    SKIP 1
ENDIF

RETURN .T.

** Changed to ADD EVENT

CASE UPPER(p_control) = "ADD EVENT"
    DO indata.prg
    RETURN .T.

** Changed to DELETE EVENT

CASE UPPER(p_control) = "DEL EVENT"
    DO deldata.prg
    RETURN .T.

ENDCASE
RETURN .T.

* Record check procedure before saving the data.

FUNCTION f_reccheck
RETURN .T.

*************************************************************************
FUNCTION: f_fieldchk
* PARAMETERS: p_objnum: The @GET object number that is calling this routine
* RETURNS: .T.
* CALLED BY: inlipids.spr
* This function checks that the data entered is ok, and returns messages
* if problems. It also sets the kl_change variable to .T. if a change
* has been made to the field.
* Field check function. Each @ GET has a VALID clause that sends in its number in
* the GET order, and checks here for any checks required before
* leaving the GET.
*************************************************************************
FUNCTION f_fieldchk
PARAMETERS p_objnum

DO CASE
CASE OBJNUM(m.ltotchol) == p_objnum
   IF patient.psex = "M"
      =f_range(m.ltotchol, "TOT CHOLM")
   ELSE
      =f_range(m.ltotchol, "TOT CHOLF")
   ENDIF
   * If changes made to the memory variable, set kl_change to .T.
   IF m.ltotchol <> lab.ltotchol
      kl_change = .T.
   ENDIF
CASE OBJNUM(m.lhdlchol) == p_objnum
   =f_range(m.lhdlchol, "HDL CHOL") -
   W
   m.lhdlchol <> lab.lhdlchol
   kl_change = .T.
ENDIF
ENDCASE
SHOW GETS
RETURN .T.

INLIPIDS.SPR

* * 06/29/95 INLIPIDS.SPR 10:19:18

* Created by Mike Chiasson
* Called by INLIPIDS.PRG
* Purpose: To display the editing panel for lipid information.
* Partially generated by the Screen Generator.

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

IF SET("TALK") = "ON"
   SET TALK OFF
   m.talkstat = "ON"
ELSE
   m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.rborder = SET("READBORDER")
SET READBORDER ON

* **************************************************
* *
* Windows Window definitions
* *
* **************************************************
* *
* Window for editing information
Appendix G: The SoftHeart Program

IF NOT WEXIST("w_edit")
    DEFINE WINDOW w_edit;
    AT 0.000, 21.250;
    SIZE 30.154,28.000;
    TITLE "Lipids";
    FONT "MS Sans Serif", 8 ;
    NOFLOAT ;
    NOCLOSE ;
    NOMINIMIZE ;
    HALFHEIGHT ;
    COLOR RGB(,,0,128,128)
ENDIF

* The control for the editing window.
IF NOT WEXIST("w_ineditcn")
    DEFINE WINDOW w_ineditcn;
    AT 0.000, 39.750;
    SIZE 30.077,28.600;
    TITLE "Controls";
    FONT "MS Sans Serif", 8 ;
    NOFLOAT ;
    NOCLOSE ;
    NOMINIMIZE ;
    HALFHEIGHT ;
    COLOR RGB(,,0,128,128)
ENDIF

* *********************************************************
* *
* INLIPIDS/Windows Screen Layout
* *
* *********************************************************

#REGION 1
IF WVISIBLE("w_edit")
    ACTIVATE WINDOW w_edit SAME
ELSE
    ACTIVATE WINDOW w_edit NOSHOW
ENDIF

@ 3.615,1.800 SAY "Total Chol";
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 6.846,2.200 SAY "HDL Chol";
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 11.000,1.800 SAY "Tot/HDL";
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 8.462,2.000 SAY "Trigly";
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 5.154,2.200 SAY "LDL Chol";
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 15.692,2.000 SAY "APOB";
    PICTURE "@!" ;
Appendix G: The SoftHeart Program

```
FONT "MS Sans Serif", 8 ;
STYLE "BT"
@ 14.000,2.000 SAY "APOA" ;
PICTURE "@!";
FONT "MS Sans Serif", 8 ;
STYLE "BT"

@ 15.615,15.600 GET m.lpapob ;
SIZE 1.000,7.167 ;
DEFAULT " " ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
PICTURE "@ZK" ;
VALID f_fieldchk(_curobj)
@ 17.692,15.800 SAY ROUND(m.ldldchol/m.lpapob,2) ;
SIZE 1.000,7.167 ;
FONT "MS Sans Serif", 8 ;
STYLE "B"
@ 20.385,12.60 GET m.llpabnd ;
PICTURE "@KM =<" ;
SIZE 1.0,1.500 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID f_fieldchk(_curobj);
MESSAGE "Use Spacebar to toggle options or type (= or <)"
@ 20.385,15.600 GET m.llpa ;
SIZE 1.000,7.167 ;
DEFAULT " " ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
PICTURE "@K" ;
VALID f_fieldchk(_curobj)
@ 22.385,15.500 GET m.lepheno ;
SIZE 1.000,7.167 ;
DEFAULT " " ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
PICTURE "@K" ;
VALID f_fieldchk(_curobj)
@ 23.9,2.200 EDIT m.llipidcomm ;
SIZE 6.000,25.400,0.000 ;
DEFAULT " " ;
FONT "MS Sans Serif", 9 ;
SCROLL;
VALID f_fieldchk(_curobj)
```

```
* *********************************************************
* * INEDITCN/Windows Screen Layout
* * *********************************************************

#REGION 2
IF WVISIBLE("w_ineditcn")
   ACTIVATE WINDOW w_ineditcn SAME
```
ELSE
    ACTIVATE WINDOW w_ineditcn NOSHOW
ENDIF

@ 7.385,7.200 GET m.kl_cont1;
    PICTURE "@*BHN /* + ;
        (LOCFILE("ico\left.bmp","BMP|ICO|PCT|ICN","Where is left?")) + "," + ;
        (LOCFILE("ico\right.bmp","BMP|ICO|PCT|ICN","Where is right?"));
    SIZE 2.615,6.800,0.800;
    DEFAULT 1;
    FONT "MS Sans Serif", 8 ;
    VALID l_vcontrol(m.kl_cont1)
@ 1.000,3.000 SAY "Event Date:";
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 2.923,4.400 SAY f_convertdate(event.edate);
    SIZE 1.000,17.333 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B"
@ 10.231,8.400 SAY "Use Arrow to" + CHR(13) + ;
    "Select Date" ;
    SIZE 2.000,12.200, 0.000 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "T"
@ 12.385,3.1 SAY "GOTO..." ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 11.20,21.600 GET m.test ;
    PICTURE "@*VN \rightarrow" ;
    SIZE 1.769,4.167,0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID l_vcontrol("NEXT", kl_nextedit)
@ 13.3,3.100 GET M.KL_nextedit ;
    PICTURE "@*" ;
    FROM K_ELISH ;
    SIZE 1.538,18.167 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID l_vcontrol("NEXT", kl_nextedit)

** Changed picture string and SIZE to,,, 10.667

@ 17.000,9.600 GET kl_cont2 ;
    PICTURE "@*VN <<Add event, + kl_delstr + "Del event, ---;\<Exit" ;
    SIZE 1.692,10.667,0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID l_vcontrol(m.kl_cont2)
@ 0.000,2.400 TO 5.615,26.600 ;
    PEN 1, 8
@ 6.462,2.400 TO 15.770,26.600 ;
    PEN 1, 8
@ 16.615,2.400 TO 28.692,26.600 ;
    PEN 1, 8

IF NOT WVISIBLE("w_ineditcn")
Appendix G: The SoftHeart Program

ACTIVATE WINDOW w_ineditcn
ENDIF
IF NOT WVISIBLE("w_edit")
    ACTIVATE WINDOW w_edit
ENDIF

* Modal used to freeze user from accessing any other windows.
READ CYCLE MODAL;
    SHOW f_show2()
RELEASE WINDOW w_edit
RELEASE WINDOW w_ineditcn
RETURN .T.

#REGION 0

SET READBORDER &rborder
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF

* ******************************
* * f_show2 Read Level Show
* * Function Origin:
* * From Platform: Windows
* * From Screen: Multiple Screens
* * Called By: READ Statement
* * Snippet Number: 1
* *
* ******************************

FUNCTION f_show2 && Read Level Show
PRIVATE currwind
STORE WOUTPUT() TO currwind

** Deleted the IF statement for disabling the DELETE EVENT button.

#REGION 1
IF SYS(2016) = "W_EDIT" OR SYS(2016) = "*
    ACTIVATE WINDOW w_edit TOP
@ 11.000,15.800 SAY ROUND(m.ltotchoyl.m.lhdlchol,2); SIZE 1.000,7.000;
    FONT "MS Sans Serif", 8;
    STYLE "B"
@ 17.692,15.800 SAY ROUND(m.lldlchol/m.lpapob,2); SIZE 1.000,7.167;
    FONT "MS Sans Serif", 8;
    STYLE "B"
ENDIF

* Show Code from screen: INEDITCN
*
#REGION 2
IF SYS(2016) = "W_INEDITCN" OR SYS(2016) = "***"
   ACTIVATE WINDOW w_ineditcn TOP
      @ 2.923,4.400 SAY f_convertdate(event.edate) ;
      SIZE 1.000,17.333, 0.000 ;
      FONT "MS Sans Serif", 8 ;
      STYLE "B"
ENDIF
IF NOT EMPTY(currwind)
   ACTIVATE WINDOW (currwind) SAME
ENDIF
16. Appendix H: Demonstrating the Ladder of Inference

There are four inference steps involved in selecting and categorizing events into the four process theories (technological imperative, organizational imperative, emergent perspective, and social technology): choice of theoretical perspectives on technology-context interaction, the categorization of the literature into the technology-context process theories, the selection of events and building of processes, and the linking of processes to the literature. Although these steps are presented sequentially, the researcher employed an iterative approach at times to encompass new research and new critical processes.

This appendix illustrates how decisions were made for each of the four inference steps for the following specific process identified in Case One:

<table>
<thead>
<tr>
<th>Description</th>
<th>Theory</th>
<th>Time</th>
</tr>
</thead>
</table>

16.1. Choosing Theoretical Perspectives

There are certain criteria for choosing theoretical perspectives for interpretive work. First, the theories must be useful and insightful categorizations of the literature and case data. Second, the perspectives must provide a method of providing cross-case comparisons, and theoretical generalization.
Appendix H: The Ladder of Inference

There are numerous process theories concerning the interaction between context and technology, as outlined in Chapter 2. Researchers in IS and technology development, however, usually discuss two common views of the technology-context relationships: the technological imperative, and the organizational or societal imperative. Most of these same researchers also discuss the inability of both theories to capture the complex nature of technology-context interaction, suggesting that new theories need to be developed (Kling & Scacchi 1982; DeSanctis & Poole 1994), or more research is required to eventually develop technological and organizational imperatives (Markus & Robey 1988).

Choosing the former strategy, I found articles discussing newer theories of context-technology interaction, using terms like adaptive structuration, emergence and institutionalization. Each suggested more or less constraints on the relationship between technology and context; emergence suggesting each technology/context interaction is unique, while adaptive structuration and structural power suggesting that technology and context both constrain and enable within certain limits. Each view either explicitly or implicitly examines the role of users and developers in determining the outcome of the technology-context interaction.

As a result, any theoretical perspective on the technology-context interaction represent on a continuum, relatively different degrees of direction, constraint and prediction about the influence of context on technology, and technology on context. Particularly important are the roles of users and developers in shaping the development and use of technology. The chosen for this thesis include the technological imperative, the organizational imperative, the emergent perspective, and social technology. These four theories are considered somewhat
exhaustive, but not mutually exclusive. Table 2-1 outlines the assumptions of each theoretical perspective. These assumptions were crucial for categorizing the information systems development (ISD) and technology development and deployment literature into the four theories.

### 16.2. Categorizing the Literature

Literature was categorized into the four perspectives based on the inference rules of Table 2-4. Although the literature could never be perfectly assigned to one of the four theories, the general direction and description of the results and opinions were used for categorization.

As examples, the following literature is discussed in relation to the example process, conflict. Each of the following articles are classified into the emergent perspective for the following reasons. Hirschheim & Newman (1991) discuss the myths governing information systems development, and suggest that the “world view” governing ISD effects the course and outcome of development. This suggests that the users and developers “world view” affects development. Turner (1987) discusses the complexity involved information system design, and how very little is understood about the processes of design. His research points to the inherent unpredictability and the “predicament” of system design; complexity. Robey, Farrow & Franz (1989) study a system development project where the complex conflict-related interaction between competing groups and the data processing department led to system failure; an emergent outcome. Newman & Noble (1990) discuss the conflict and political forms of user involvement, suggesting that these are more descriptive of user-analyst relations than learning. As a result, “political forms of involvement” lead potentially to a “garbage can”
model of development, where random collisions and events produces an indeterminate and complex development path. Finally, Thomas (1994) develops a *process-political* model of technology development in the manufacturing field, placing his work between social technology and the emergent perspective. His discussion of innovations generating from process manufacturing engineers in large organizations looking for prestige and enhanced status, and the reaction of research and development power brokers in the organizations to capture and transform the innovation to meet their power needs, demonstrates a "political" form and complex path of technological development.

### 16.3. Events and Processes

*Events* are discrete points in time (usually during a day) that appeared during the case study. *Processes* are the more stable and structured interaction between technology and context over time, illustrated by events.

Although each event was not directly categorized into a process during data analysis, all events in both cases were analyzed and used to develop the processes, including their first appearance. In addition, although the sixty eight processes cannot be considered exhaustive of all processes in the case studies, they do represent critical and illustrative processes.

There were basically two types of processes; those involving a smaller number events over a concentrated period of time, and those involving a larger number of events scattered over longer periods of time. The *conflict* process is an example of the former. The following shows the events that indicated a conflict process:
<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan. 30, 1995</td>
<td>Physician warns about conflict at Clinic One and admires the developer’s willingness to get involved. The physicians are worried about the coordinator’s poor data entry into EasySys. The developer begins to feel the animosity between him and the coordinator.</td>
</tr>
<tr>
<td>Feb. 22, 1995</td>
<td>A meeting was held with Hospital Two’s ethics advisor. He mentioned that the chart review form was created by the coordinator in the last couple of weeks to handle demands for data querying from EasySys. He is surprised that she is moving so quickly on enforcing use this form.</td>
</tr>
<tr>
<td>Feb. 23, 1995</td>
<td>A prototype demonstration with the coordinator led to derision and hostility. The coordinator demanded that SoftHeart look identical to EasySys; a system developed by her and a programmer. A clerk mentioned that the coordinator does not like the developer’s work in the clinic; causing too much change.</td>
</tr>
<tr>
<td>Mar 2, 1995</td>
<td>The coordinator produces a new chart review form required by Clinic One to authorize the developer’s continuation with the study.</td>
</tr>
<tr>
<td>Mar 21, 1995</td>
<td>Discussion with a physician reveals a long history of conflict between the coordinator and other staff members. The director’s lack of control over the coordinator has created an administrative fiefdom.</td>
</tr>
</tbody>
</table>

Determining the exact starting time of processes were difficult, given the developer’s discovery of conflict between the coordinator, the research group, and the physicians in early September. Nevertheless, blatant signs of conflict began to appear in January, as the developer began redeveloping EasySys functionality within SoftHeart.

These events were then categorized as a conflict process within the conflict erupts stage of development, as the schism between the research group and the coordinator reappeared, with the developer of SoftHeart unknowingly caught in the middle.

### 16.4. Linking Processes to the Literature

Inference rules from processes to the literature are provided in footnotes in Chapters 4 and 5. In this example, the conflict process was linked with research by Turner (1987), Turner (1987), Robey, Farrow & Franz (1989), Hirschheim & Newman (1990), Newman & Noble (1990) and Thomas (1994). As stated in Chapter 4, goals were ill defined, and the
means to those goals were fuzzy and obscured by the developer’s inability to distinguish legitimate demands from user resistance during this period. Design complexity and new issues emerged (Turner 1987). Unresolved issues of control and direction in the clinic amongst the clinic personnel continued to hamper sustained development (Robey, Farrow & Franz 1989). Conflict and learning about requirements, opportunities and structural conflict occurred during this process (Newman & Noble 1990). Finally, a political struggle emerged with the over the course of SoftHeart future direction and “look-and-feel” (Thomas 1994).