BUSINESS-DRIVEN INFORMATION SYSTEMS CHANGE: HOW COMPANIES ANTICIPATE, PREPARE FOR, AND RESPOND TO THE NEED FOR CHANGE

by

JUSTIN DANIEL COCHRAN

(Under the Direction of Dale L. Goodhue)

ABSTRACT

Organizations are continually faced with the reality that they must change their information systems (IS) to stay competitive in the marketplace. While previous research has focused on the role of IS in contributing to organizational agility by enhancing abilities such as data collection and decision making, less research has examined the case where organizational changes require changes to the IS. This research creates a framework which links the need for organizational change to the need for information systems change. The framework further connects the ability to change the IS, or IS "response ability", to organizational agility. IS "response ability" is affected by two primary constructs: business-driven IS change and the application characteristics relevant to the change.

To examine the constructs and relationships between business-driven IS change, application characteristics relevant to change, and IS response ability, a series of exploratory semi-structured interviews were conducted at five organizations. These interviews resulted in evidence which confirmed and extended the model initially developed to guide this study. In addition, the study also resulted in new findings – organizations use IS change anticipation strategies and IS change anticipation tactics to prepare for future IS change needs.

The major contributions of this study are elaboration of the relationships between business-driven IS change, application characteristics, and IS response ability. In addition, this study discusses specifically which characteristics of applications have effects on the ability to make IS change. The second major contribution of this study is that organizations choose various anticipation strategies and anticipation tactics in preparation for change.

The results of this study have implications for numerous parties. Organizations should see greater evidence for the connection between information systems and agility, perhaps in a way they had previously not considered. IT leadership should recognize a need for greater emphasis on predicting business-driven IS change and, accordingly, the necessity to be more involved with the business side of the organization. Developers should understand the range of options at their disposal for preparing applications for IS change. All parties should recognize that applications can no longer just be programmed to function – they must be programmed for change.

INDEX WORDS: Information Systems Change, Anticipation Strategy, Anticipation Tactics, Prediction, Response Ability, Application Characteristics Relevant to Change, Organizational Agility, Flexibility

BUSINES-DRIVEN INFORMATION SYSTEMS CHANGE: HOW COMPANIES ANTICIPATE, PREPARE FOR, AND RESPOND TO THE NEED FOR CHANGE

by

JUSTIN DANIEL COCHRAN

B.S.M.E., Auburn University, 1998

M.S.M.E., Auburn University, 2000

A Dissertation Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2008

BUSINESS-DRIVEN INFORMATION SYSTEMS CHANGE: HOW COMPANIES ANTICIPATE, PREPARE FOR, AND RESPOND TO THE NEED FOR CHANGE

by

JUSTIN DANIEL COCHRAN

Major Professor:	Dale L. Goodhue
Committee:	Robert Bostrom Marie-Claude Boudreau Dave Chatterjee

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
August 2008

To My Family

ACKNOWLEDGEMENTS

I would like to thank my advisor and friend, Dale Goodhue, for his support of my work. Without his guidance, his knowledge, and mostly his patience, this research would not have been successful. Dale has shared a good portion of the last few years going along on this ride with me, and while it meandered, I can only hope it was at least fractionally as memorable and meaningful to him as it has been to me.

I would also like to thank the rest of my committee: Bob Bostrom, Marie-Claude Boudreau, and Dave Chatterjee. They have each lent a critical eye to my research on a number of occasions. Just as important, they have each offered encouraging words, which motivated me to keep thinking and keep working. I could not imagine a better committee.

I would like to thank the faculty in the MIS department. They create an atmosphere of positive criticism, which results in both higher quality analysis and motivation. This has been essential for my growth as a young academician. I would also like to thank Mark Huber and Craig Piercy for advising me on teaching issues, providing assistance with new classes, and for going to lunch with me frequently.

I would also like to thank my peers in the MIS and Management PhD programs. Iris Junglas, Dominic Thomas, Chris Kadlec, Hui Wang, Martina Greiner, Clay Williams, Donald Wynn, Paul York, and Ashley Davis in the MIS department, and Jill Brown, Peter Gianiodis, Ken Park, Kareem Shabana, Jeff Street, Marcus Butts, Tom Will, and Laura Stanley in the Management department, have all helped make my experience enjoyable.

I would especially like to thank Saurabh Gupta and Stacy Campbell. They have been great colleagues and even better friends in the past few years. They have served as sounding boards for countless research ideas, both good and bad, and reviewers on many different research scraps. They have continually encouraged me through struggles in the program. Saurabh had the fortune of sharing an office with me, and consequently, was forced to endure chalkboard drawings of models and discussions about all things. Stacy was instrumental in setting up opportunities for me to interview industry practitioners, and without her, my list of research would be extremely short.

I would like to thank the organizations that took a leap of faith with me on this project. There was little I could offer in return to them, but they allowed me access to pick the brains of their personnel for my research. Without their kindness, this research would not be complete. My hope is that what is contained here will be of some use to them in the future.

Finally, I would like to thank my family. My parents, my brother, and my grandmother have all supported me throughout my life and they should get some satisfaction from my completion of this program. They have provided me with a solid footing to stand on and motivation to keep going.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1 INTRODUCTION

1.1 INVESTIGATION BACKGROUND AND IMPORTANCE

For many organizations, information systems (IS) serve as the backbone for conducting the necessary business processes for their operations. Through information systems, these organizations can realize greater productivity (Hitt et al. 1996) through reduced human resource participation and error, greater efficiency, and greater speed in conducting the steps of the processes. In fact, organizations can become so dependent on their IS that they would find themselves operationally crippled if the IS were shut down. Yet, while information systems have led to great operational efficiencies in many cases, the rules-based nature of IS often "hard wires" organizations to the point that the systems and, potentially the organization, become unable to change as needed. This constant struggle between the efficiencies enabled by IS, its rigid tendencies, and the need for business change continuously challenges IS personnel. To this day, researchers and practitioners alike are still trying to understand the balance between these forces.

The impacts of Sarbanes-Oxley legislation on the information systems of public companies provide a contemporary example of this struggle. In an effort to better control reporting and increase transparency of financial and accounting practices, the U.S. federal government passed the Sarbanes-Oxley Act (SOX) in 2002 (Kaarst-Brown et al. 2005). Given the interwoven nature of financial and accounting processes with information systems in organizations, this legislation resulted significant changes to the IS, such as "reporting content, timeliness, retention and destruction policy, detailed documentation, and integration of information from manual and automated processes" (Kaarst-Brown et al. 2005), often requiring

millions of dollars (Insight 2004). In one report, over 45% of organizations surveyed are rewriting applications, 42% are developing new applications, and about 30% are buying new applications in response to SOX requirements. Additionally, 34% of organizations surveyed state that compliance with SOX regulations is decreasing productivity (Violino 2003). These changes to the IS vary in difficulty because of the existing IS infrastructure, which includes the 1) hardware and operating systems, 2) network and telecommunication technologies, 3) key data, and 4) core data-processing applications (Duncan 1995).

This example, originating from legislative changes, illustrates how critical it can be for the organization's IS infrastructure to be malleable. As a consequence, organizations are considering options for their IS infrastructure that are more easily changed as needed, such as service-oriented architectures. While IS departments and the business community are surviving through the IS changes required by SOX regulations, organizations will continue to struggle with making IS changes because they are continually developing, redesigning, and eliminating business processes in order to better compete in the marketplace.

To better understand how organizations respond to change, this research takes a closer look at both the IS infrastructure characteristics and IS change requests from the business. In particular, we are interested in the application infrastructure, including the key applications and data, because of their importance to conducting the organization's processes. By studying business-driven IS changes and their interaction with the application infrastructure, this research provides a more complete understanding of how the IS can respond to change needs.

1.2 RESEARCH QUESTIONS

At the highest level, this research is concerned with understanding the ability to change the existing IS infrastructure[1], specifically the application infrastructure, in response to business needs. For this research, we are most interested in the key applications and associated data components of the application infrastructure because of their connection to the execution of the business processes. After information systems are implemented, changes to the IS are not easily enacted, despite the wishes of business managers and the best intentions of IS personnel. This is primarily due to the inescapable presence of the existing IS infrastructure, and the accompanying specific and rules-based nature of that software. In other words, software has to be designed to include every eventuality that may crash its operation. Meanwhile, IS developers are trying to better prepare for the inevitability of post-implementation change. To do this, they are developing reconfigurable systems, modular architectures, and interoperability standards.

To better understand the challenges IS personnel are facing, we first focus on the types of IS changes that they encounter. Because IS changes differ in their impact on the infrastructure, it is important to construct a typology or range of change needs. This will allow future research and system developers to better understand how to design systems for change, the difficulties associated with particular types of change, and ways to address the change needs. Accordingly, the first research question of this study is: *What are the types of change required of the IS by the business?*

Additionally, we need to better understand the constraints to change. Given that the application infrastructure is both the environment for change and a likely constraint, we need to examine the aspects of the applications that make it difficult to change (or in some cases, easy).

[1] This research will focus on the application infrastructure as part of the IS infrastructure for reasons that will be discussed later. In general, the discussion of infrastructure will begin broadly with the IS infrastructure for conceptual development and narrow to particular aspects of the software infrastructure as we get closer to measurement.

The ease or difficulty of change can be influenced by structural factors and design choices, as well as personnel factors. Hence, the second research question of this study is: *What are the application characteristics that impact the ability to change?*

Thirdly, since the difficulty of change in this environment is a product of both the IS change itself and the application infrastructure, it is important to understand the interaction between the two. This interaction dictates certain response options that can vary from one situation to the next. The third research question of this study is: *What are the response options of the IS, given a particular IS change need and a particular software infrastructure?*

To examine these questions, academic and practitioner research in flexibility and agility are utilized due to their relevance to IS change. Examination of this existing research has suggested two research gaps that are addressed with this current research. First, agility research has primarily looked at the contribution of IS as it pertains to organization-level benefits. Additionally, the discussion of IS in agility research focuses on the ability of IS to assist in sensing agility, which consequently improves knowledge and decision-making, allowing the organization to respond appropriately (Overby et al. 2006; Sambamurthy et al. 2003). Less research has explored the impact of organizational agility needs on the information systems, and the ability of the IS to accommodate those required changes.

Second, previous IS flexibility research has implied that flexibility is a property of the system (Broadbent et al. 1999; Byrd et al. 2001; Duncan 1995). In other words, flexibility research has examined the circumstance where the required change to the system was expected and planned into the system. This research examines additional IS changes, where the change was not anticipated or designed into the system, and consequently may be more difficult.

1.3 RESEARCH METHODS

As noted, this research addresses a couple of gaps in existing research while answering the three primary research questions. However, the literature bases are insufficient for conducting a quantitative study effectively given the lack of direct applicability to this research and the lack of validated measurement tools. Therefore, this study uses a qualitative approach to explore the research questions as well as the context so that future research will have a broader footing to develop upon.

Twenty-seven semi-structured interview sessions were conducted with thirty-one informants in various IS roles within five organizations. Of these, twenty-one of the interviews were selected for analysis due to their relevance to the research topics. The remaining six interviews were either not recorded or were not relevant to this study. The five organizations represented five different industries including retail, insurance, utility, software, and banking. Within four of the organizations, informants included senior managers, analysts, project managers, and developers. The combination of different roles and different organizations yielded many different perspectives for comparison during data analysis.

Data analysis used an initial coding template based on concepts from the literature review. The template was iteratively modified as interviews were analyzed by multiple coders. Once the coding template was relatively stable, two coders independently coded and subsequently discussed the interviews. This continued until independent coding yielded an interrater reliability coefficient of 0.845.

The research method used in this study allowed for validation of the initial conceptual model developed prior to interviews. Additionally, the interview method also allowed for significant new findings to emerge. These new findings represent additional key contributions of this study.

1.4 CONTRIBUTIONS

This research makes a number of key contributions to the IS field, both theoretical and practitioner. From a theoretical perspective, this research looks at an alternate perspective of the role of IS in achieving organizational agility. Prior research has looked at the role of IS as a tool to assist in sensing environmental change for the organization and for providing better information for decision making for organizational response (Sambamurthy et al. 2003). This research examines cases where part of the organizational agility response requires changes to the information systems of the organization. In other words, organizational agility may require changes to the IS, and the ability to change the IS can, in turn, affect organizational agility.

Additionally, this research frames the ability to change the information systems as the IS Response Ability. This frame emphasizes that the IS response is comprised of both the particular IS change need and the application characteristics relevant to that change. While other research has discussed the inertial effects of IS (Overby et al. 2006; van Oosterhout et al. 2006), no known research has yet detailed what characteristics of the IS affect whether change is easy of difficult.

A third theoretical contribution of this research is the exploration of the anticipation of IS change by organizations and the resulting measures taken to prepare for potential IS change. This resulted in a new model that describes a number of different factors that affect the design of IS based on the recognition that applications will need to be changed after they are implemented.

From a practitioner perspective, organizations have a framework for understanding and explaining the common difficulty of making business-driven IS changes. As a result, IS departments can illustrate their role in achieving organizational agility, and consequently, the importance of resources to enable them to make the necessary changes. Furthermore, the frequent difficultly of making IS changes can be traced back to design decisions made prior to

implementation. This indicates that the speed at which IS changes are made is determined in part well before the change request is placed. These design decisions can be tied to anticipation strategies, which emphasizes the need for thorough planning of applications as well as closer ties between the business and the IS department.

1.5 STRUCTURE OF THIS DOCUMENT

Chapter 2 begins by defining the context of this study. Specifically, research has indicated that IS can play multiple roles in achieving organizational agility (Chen 2004; Overby et al. 2006; Sambamurthy et al. 2003). Among other reasons, we have chosen to research the responding role of IS due to the constant pressures businesses are under to change and its current lack of research.

Subsequently, a guiding model is introduced describing the components and suspected relationships between business changes affecting the IS, the resulting IS changes, the application characteristics relevant to the change, and the IS response types. Each of these constructs are expanded and elaborated upon afterwards. Given their connection to change, the pertinent literature in flexibility and agility is used to help construct the ideas in the model.

Chapter 3 describes the qualitative design that is used to conduct this study. As mentioned, this area of research is underdeveloped and lacks suitable quantitative measures at this point. Therefore, semi-structured interviews are used for gathering specific examples of IS change and the surrounding context, which is subsequently analyzed.

Chapter 4 describes the findings from the analysis of the interviews. First, the findings related to the research questions are discussed in detail. Next, new findings related to the anticipation of IS change are described and a new model is presented.

Chapter 5 presents a synthesis of the models developed in this work, the findings of the study, the limitations, contributions to theory and practice, as well as future research directions.

CHAPTER 2 LITERATURE REVIEW AND CONCEPTS

"Every request for change gets to be a giant project that takes forever to complete and costs a fortune" (Allen 1982) discussing inflexible systems.

2.1 RESEARCH FOUNDATIONS

At the highest level, this research is concerned with understanding the ability to change the existing IS infrastructure, specifically the application infrastructure, in response to business needs. Accordingly, this research uses a literature base related to change and to IT flexibility as its starting point. The following sections describe the relevant research and how this research fits into the gaps remaining.

2.1.1 Agility Research

Previous research has theorized the role of information technology (IT)[2] in enhancing agility for businesses at the organizational level (Overby et al. 2006; Sambamurthy et al. 2003), but the primary focus has been on the ability of IT to help organizations detect threats and opportunities in their environment. In other words, how does IT help the organization collect and manage information and knowledge for better decision making? More specifically, they conceive of IT as a digital options generator, where digital options are organization capabilities enabled by IT. In this view, IT enables more information to be captured, processed, and utilized, which improves the ability of the organization to sense and respond to opportunities (Sambamurthy et al. 2003). Additionally, IT can have a direct or indirect effect on organizational agility. Directly, IT can contribute to organizational agility through IT-enabled

[2] In this document, the abbreviation IT is intended to refer to the technology components, while IS refers to the broader information system, which includes technologies, people, processes, and structures (Bostrom et al. 1978).

initiatives, such as e-commerce, customer relationship management, and supply chain management. Indirectly, IT improves process and knowledge reach and richness, creating digital options which may help organizations detect and respond to more opportunities effectively (Overby et al. 2006).

This research fits under the broad umbrella created by this above research on organizational agility enabled by IT. More specifically, previous research emphasized the ability of IT to help improve decision making, thereby yielding a better agility response. This response may or may not include changes to existing applications responsible for conducting the business processes. This research examines the situations where there are changes required to the existing information systems as part of a larger organizational change. This research area is particularly important because organizational agility is, in part, based on speed and surprise (Sambamurthy et al. 2003), yet, as described in the following section, changing information systems is quite often not a trivial or speedy process.

2.1.2 IT Flexibility Research

While the difficulty of changing information systems to meet business needs is not a new development in the field, a suitable theoretical model to guide research regarding change to the IS infrastructure does not yet exist. The closest research area related to the ability to change IT in response to business needs is research on IT flexibility. For many years, authors have noted the importance of designing flexible information systems so that change is easier. In fact, systems should be structured to accommodate changing information requirements, including those that are not anticipated (Allen et al. 1991).

However, this is often easier said than done. Duncan (1995) notes that inflexibility is the "difficulty…with users' demands that require systems to do things they were not designed to

do." She notes an example from an informant where in one legacy system, "sixteen or seventeen business processes could be embedded in the centralized system" and a change to one process could impact all of the others. To combat the inflexibility associated with IS, companies are developing modular applications and data, which involves "isolating and standardizing as many business and systems processes as possible." (Duncan 1995)

Broadbent et al. (1999) noted that incompatible systems, inconsistent data models, systems based on old or inappropriate processes, and data and systems built to serve local and functional needs may constrain change to business processes. In other cases, however, IT may enable change for the organization by allowing companies to operate in new and more efficient ways (Broadbent et al. 1999). More recently, research has argued that IT can be both an enabler and inhibitor for achieving organizational agility based on its difficulty to change. Van Oosterhout (2006) found that embedded business processes and "complex nests of links between applications" made change to systems difficult. However, organizations with simple system architectures or few legacy applications made changes more easily (van Oosterhout et al. 2006).

This research asserts that more research is needed in this area because previous IT flexibility research implies that flexibility is a property of the system (Broadbent et al. 1999; Byrd et al. 2001; Duncan 1995). If all possible IS changes were described as a set, then this characterization is reasonable when the changes made to the system are within the subset of anticipated changes. However, this implication underemphasizes unforeseen changes that fall outside of the subset of anticipated changes incorporated in the original system design, and consequently cannot be easily accommodated, such as those associated with organizational agility maneuvers. In other words, previous IT flexibility research is not sufficient for describing the full range of IS changes required by modern businesses, especially those unanticipated

changes often brought about by business changes. This research extends and reconceptualizes previous research on IS change to broaden the view to those changes that are unanticipated, and frequently not easily accommodated.

To begin understanding how the business needs for IS change are met, we need to understand at least four different areas. First, we need to understand the business environment of the IS infrastructure, namely the business changes that require changes to the organization's information systems. Second, we need to understand the general IS changes that are commonly prompted by business change. Third, we need to understand any constraints to change, such as the structure, types, and characteristics of the existing IS infrastructure and those responsible for enacting the changes. Finally, we need to understand how the particular system change needs and the properties of the IS interact to create particular response options. These four areas will comprise the general model guiding this research.

2.1.3 Structure of this Chapter

This chapter continues by describing how the need for organizational agility often impacts the information systems. Discussion is devoted to this topic because of the multiple roles that IS can play in achieving organizational agility. Because of these multiple roles of IS in agility, it is important to define clearly the context of this study to illustrate where this study fits with other views. Next, the constructs that are investigated in this study are described: the business changes impacting the IS, the business-driven IS changes, the application characteristics relevant to change, and the IS response types for the information systems.

2.2 RELATIONSHIP OF ORGANIZATIONAL AGILITY TO INFORMATION SYSTEMS

Information systems research has touted the importance of information systems for achieving organizational agility (Overby et al. 2006; Sambamurthy et al. 2003). Dove has

depicted organizational agility as the product of "knowledge management" and "response ability". In his view, knowledge management is broadly defined as "the intellectual ability to find appropriate things to act on", while response ability is simply "the ability to act" (Dove 2001). This view of the major components of organizational agility is consistent with other authors' views that agility is comprised of the ability to sense opportunities (sensing agility) and the ability to react to those opportunities (responding agility) (Chen 2004; Overby et al. 2006; Sambamurthy et al. 2003). While many organizations recognize the need for organizational agility, as well as the potential of IS to support that need, it is important to understand specifically where IS can impact organizational agility. Figure 2-1 represents a breakdown of the relationship between IS and organizational agility (a variation of Chen, forthcoming).

Information systems support both of these components of organizational agility – finding opportunities (sensing agility) and responding to opportunities (responding agility). While this distinction between the two components has been made at the organizational level, published information systems research has yet to leverage the distinction at the IS level to better define the role of IS in achieving agility. In fact, by using the term agility in multiple ways to describe information systems, the research is arbitrarily lumped together, disguising that IS supports sensing and responding agility in two different ways.

Much of the current research on agility is focused on the sensing agility side of the model, perhaps because of the expected impacts of business intelligence and knowledge management on organizational agility. This research, however, focuses on the responding agility side of the model, *where information systems must change due to business demands*.

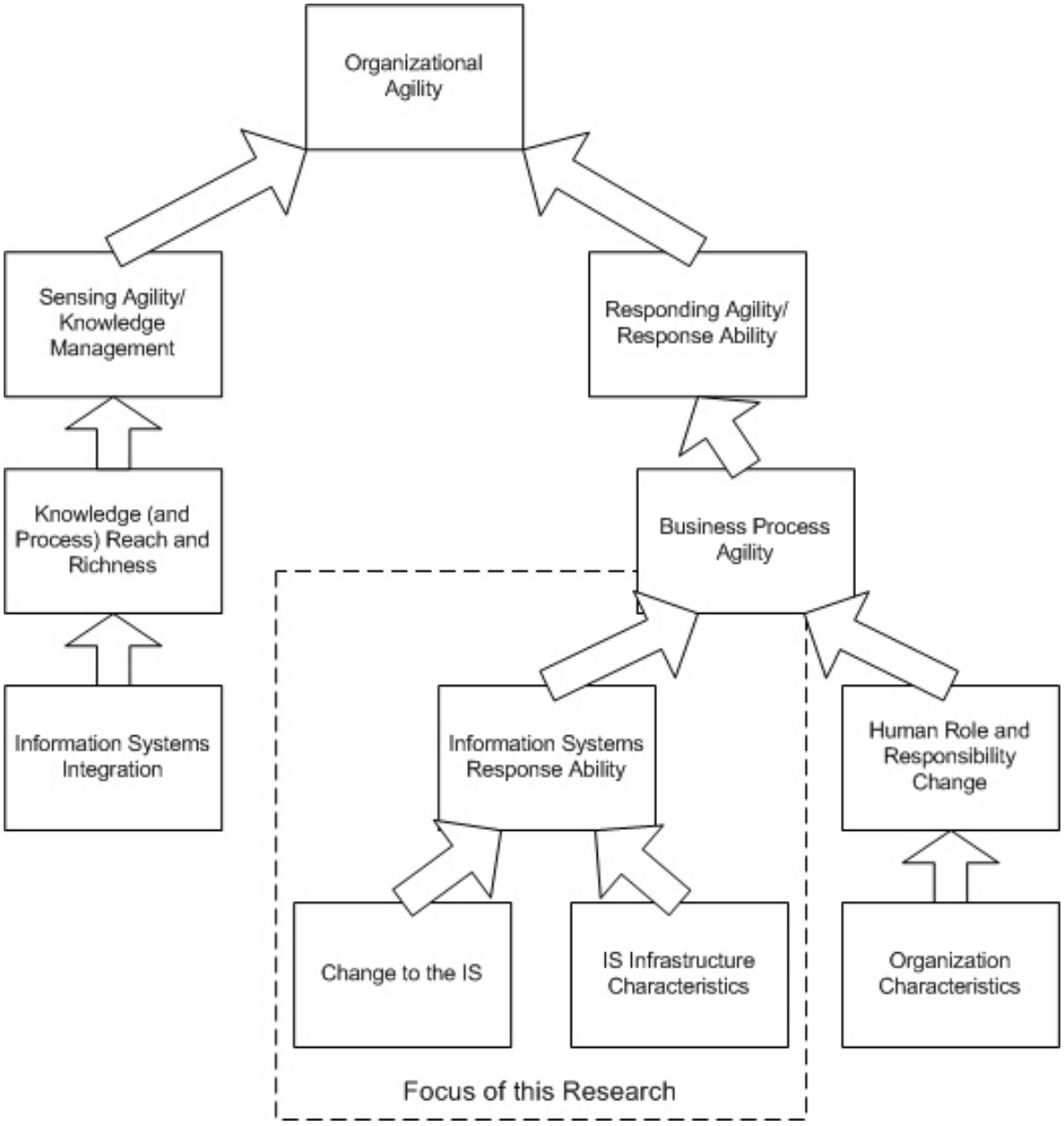

Figure 2-1: Conceptual Model of the Role of IS in Achieving Organizational Agility

2.2.1 Sensing Agility

According to some authors, IT creates options for organizational agility by extending the knowledge (and process) reach and richness of the organization (Overby et al. 2006; Sambamurthy et al. 2003). This view is consistent with the role of IT in the information processing view of the organization (Galbraith 1974; Tushman et al. 1978). In other words, by

harnessing the capabilities of IT, such as integration (among other things), the decision makers can gain access to relevant information faster, and make faster and better informed decisions, increasing the potential for achieving overall agility (Chen 2004).

2.2.2 Responding Agility

One common area that is impacted during organizational agility responses is business processes. For example, new shipping processes, modified sales processes, or removal of redundant processes may be candidates for responding to sensed opportunities. In other words, *business process agility*, or adding, removing, or changing business processes, is one key area that organizations must often attend to if they are to be able to respond to opportunities.

In organizations, there are two primary means for executing the business processes: people and information systems. Most likely, there is a combination of people and systems for most processes. Therefore, changing the business processes could require a change in the steps of the process, a change in the information systems, a change to the human roles and responsibilities, or some combination of the three. In practice, each of these components provide some resistance to change. Since this research is primarily concerned with information systems, we will concentrate on the IS while acknowledging that other factors can impact business process agility.

As one author noted, "much of the business value of IS stems from its complementarities with business processes" (Barua et al. 1995). This indicates the interwoven nature of business processes and information systems in modern organizations. Therefore, when changing business processes, one must typically modify the IS, which may or may not be easily changed depending on the IS infrastructure characteristics.

2.2.3 The Bi-directional Relationship between Organizational Agility and IS Response Ability

In the previous sections, we have superficially described how organizational agility can be supported by the information systems of the organization. At this point, we need to further elaborate on the relationship between organizational agility and information systems. Specifically, information systems, due to their rigid nature, provide resistance to change, which impacts the ability for the organization to be agile. So there is a bi-directional relationship between organizational agility and information systems where each impacts the other.

One way to examine the relationships is illustrated in Figure 2-2. There are two sides to achieving agility, the need and the ability to change. To reiterate previous statements, opportunities or problems create an organizational need for change. The source of this need may be internal, such as the need to streamline a process for greater efficiency, or external, such as competitive threats or regulatory changes. Quite often, the result is a need for business process change. Given the aforementioned interwoven nature of business processes and IS in modern organizations, the need for business process change often creates a need for IS change.

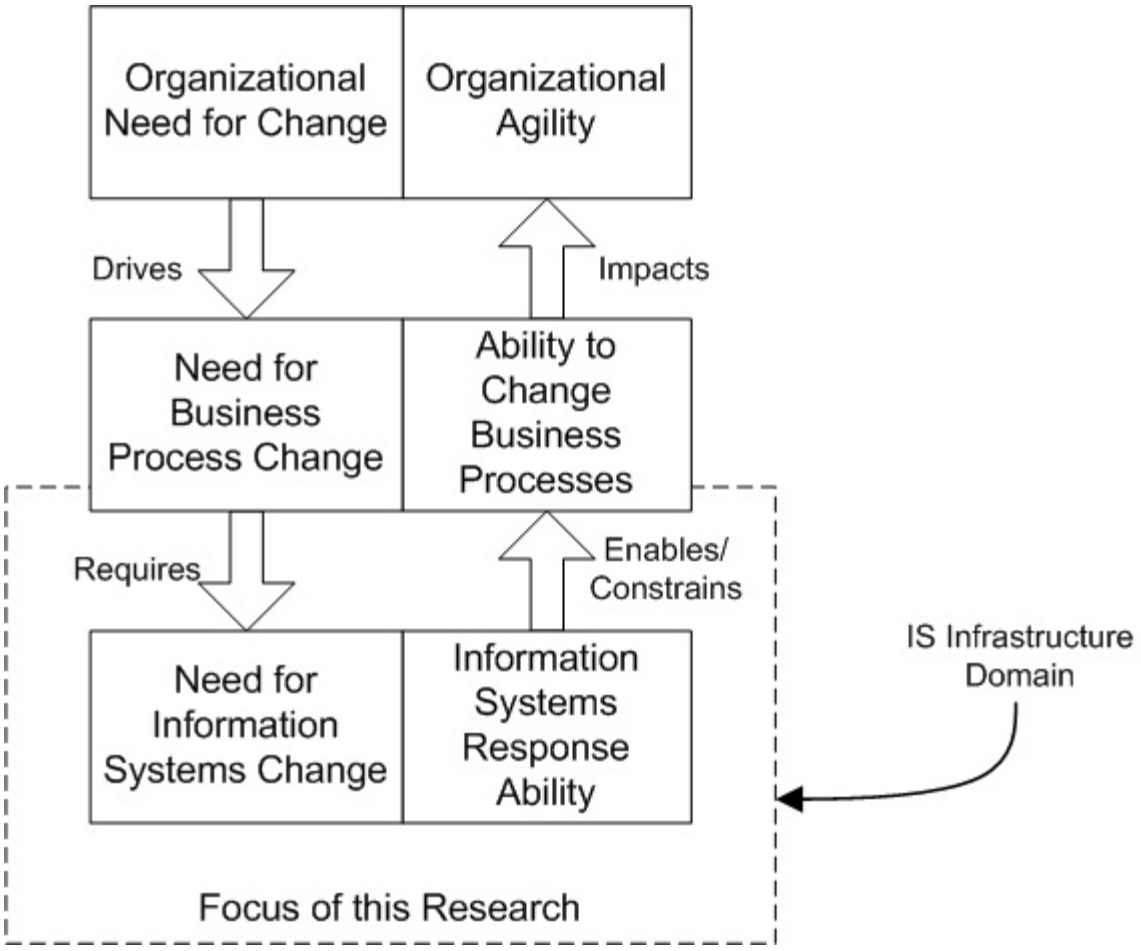

Figure 2-2: Bi-directional Relationship between Organizational Agility and IS Response Ability

IS change, however, is subject to the characteristics of the system, such as the level of integration, modularity, and system type. The ability to change the IS infrastructure as needed is what we term as *information systems response ability*. IS response ability can be viewed as the difficulty of meeting the need for IS change. As we will illustrate in later sections, IS response ability is affected by both application characteristics and the required IS change.

Depending on the difficulty of changing the IS, the IS can have a major impact on the ability to change the business processes, which in turn affects organizational agility. This view of the relationships between organizational agility and IS infrastructure illustrates the double-

edged sword of IS. Organizations find great benefits in operational efficiency, and perhaps sensing opportunities, while the systems are static. However, when there is a need for IS change, whether it comes from sensed opportunities, or any number of other sources, the inability of an IS to respond quickly can have hugely detrimental impacts on organizational agility.

2.2.4 The IS Infrastructure Domain

The discussion to this point is not intended to indicate that the only means of achieving organizational agility is through changing business processes, and subsequently, the information systems. Nor do we suggest that the only resistance to change in the organization is the information systems. It is simply to emphasize that a common outcome of organizational agility maneuvers is change to the information systems. Also, information systems can potentially be poor at responding to change needs, eventually impacting organizational agility.

In fact, the components of an organizational agility maneuver could be so varied that information systems change might only be one aspect. Therefore, for the purposes of practicality and depth, we are defining the *IS Infrastructure Domain* as the area contained within the boundary of direct influence on the IS. Within the IS infrastructure domain is the IS infrastructure itself (including the hardware, software, networking, and data components), the IS staff and leadership. Accordingly, the customers of the IS and others who can influence change to the system (such as vendors and regulators) sit outside of the IS infrastructure domain. This will serve as the context and boundary of this research.

By delineating the IS infrastructure domain, we benefit in a few ways. First, it allows us to examine the interface between the business and the information systems rather than the broader organization environment. Therefore, we are interested in the change requests that the IS department receives, with less emphasis on the origin of the change. For example, a

corporation's competitor may begin offering shipments of products direct from the manufacturer. While the organization may respond in a number of ways across the organization to this environmental change, we are most interested in what requests come to the IS department, and the ability of the application infrastructure to respond.

Second, we can discuss more clearly what is meant by terms such as flexibility and agility within this context, as well as concepts such as environmental diversity. Environmental diversity describes the variability, variety, or complexity of the environment (de Groote 1994) and may serve as one starting point for evaluating change types to the IS. For our purposes, we can examine the change requests coming into the IS infrastructure domain from the environment in terms of environmental diversity, or whether the possibility of the IS change was foreseen or unforeseen. Other distinguishing characteristics of the changes may be urgency, degree to which change is needed in a "backbone" system or peripheral system, and degree to which workarounds are available.

2.3 CONCEPTUAL MODEL FOR THIS RESEARCH

Based on the previous discussion, we have constructed a conceptual model (Figure 2-3) to drive this research. It is introduced here to frame the elaboration on the various components of the model that follow. In general, the model argues that particular business changes drive IS changes, which when paired with particular application characteristics, yield three general response types for the system. These response types each have varying effects on the performance[3], at both the IS department level and the organizational level. In addition, previously anticipated changes to the IS affect the design and assembly of the applications comprising the infrastructure, which may or may not allow for particular responses when the IS change is necessary.

[3] This is a good candidate for future research, but for now, is outside of the scope of this research.

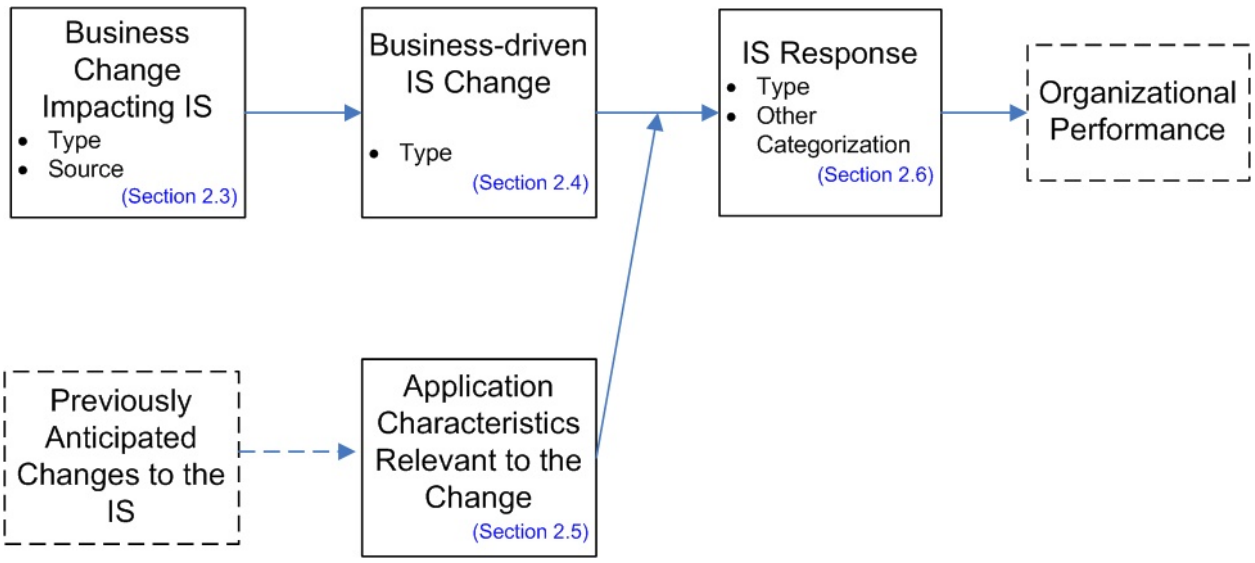

Figure 2-3: Initial Conceptual Model

In the following sections, each of the major constructs will be discussed within the context of this research.

2.4 BUSINESS CHANGES IMPACTING THE IS

There can be a wide variety of changes (new product development, restructuring, mergers, etc.) that a business undertakes for a wide range of reasons (financial, performance, competitive, etc.). Inevitably, many of these business changes result in changes to the IS (see Figure 2-2). Some examples of the types of changes may be business process improvement, regulatory or legal changes, mergers and acquisitions, or reporting changes. To reiterate, the business changes impacting IS can be a product of higher level organizational agility requirements.

Two examples from this study, both related to natural disasters, illustrate how business changes impact the IS of organizations. When hurricanes hit the coastal areas in the southeast US, InsuranceCo needs to be able to efficiently route insurance adjustors to customers' homes.

The IS department had to make changes to their applications to capture county data when policies were written, allow for policies to be searched for by county, and integrate a mapping feature to help adjustors with route planning. Under typical circumstances, these capabilities would have been nice to have. But under the pressure of hurricane damage, this became a necessity which drove IS changes.

PowerCo also has to deal with hurricane damage as well. When they need to repair power lines and related equipment, they must mobilize lots of personnel and equipment, often to other states. PowerCo needed better logistics planning for cases where damage was widespread and resource demands were high. This business requirement led to the acquisition of a logistics planning tool, which they integrated with their existing systems.

2.5 BUSINESS-DRIVEN IS CHANGE

Business-driven IS changes are a primary interest of this study. The focus of this research is on business-driven changes, rather than technology driven changes (such as techincal upgrades, patches, etc.) because of our interest in changes that come from demands outside of the IS departments. One angle to view business-driven IS changes is as high-level changes, independent of particular application implications. For example, one such high level IS change might be “process change” rather than “change to an ERP process”. By focusing on the high level IS changes, we can better separate the change type from the application characteristics (see section 2.6). This is important because of the influence of particular application types and designs have on the ability of IS departments to make changes to the systems.

In the previous example, InsuranceCo had a number of different business-driven IS changes to make because of business needs. These IS changes include collecting data (capturing

county), changing a process (change of search function), and process integration (adding mapping features to existing reports for adjustors).

In addition to the typology of IS changes, this research examines whether particular IS changes were anticipated. Conceptually, it is expected that changes that were foreseen should be easier to change when those changes are required[4].

2.6 APPLICATION CHARACTERISTICS RELEVANT TO THE CHANGE

Information technology infrastructure has been defined as "a set of shared, tangible IT resources that provide a foundation to enable present and future business applications" The four primary components are 1) hardware and operating systems, 2) network and telecommunication technologies, 3) key data, and 4) core data-processing applications. (Duncan 1995) In practice, the IT infrastructure may contain many different components, such as mainframes and servers, databases, enterprise resource planning systems, wired and wireless networks, and so on. For this study, we are primarily concerned with the application infrastructure[5], specifically the *business applications and data responsible for conducting the business processes of the organization.*

This study narrows the focus to the business applications and data responsible for conducting the business processes of the organization for a couple of reasons. First, those applications and data are expected to be impacted when business-driven IS changes are necessary. Hardware, on the other hand, may not change at all for many business-driven IS changes. Second, for practical purposes, it would be difficult to focus on the entire IT

[4] It is possible that changes can be foreseen, but never planned into the infrastructure. Some reasons this may occur are resource and time limitations, low probability of occurrence, or difficulty in implementing.
[5] Conceptually, we are concerned with the application infrastructure. However, as described later, data collection will narrow the focus further to particular systems that respondents are familiar with.

infrastructure because of the many different types and variations found within large organizations.

Additionally, studies have also grouped IS personnel into the infrastructure discussion (e.g. (Broadbent et al. 1999; Byrd et al. 2001; Henderson et al. 1994; Lee et al. 2005)). This changes the view of IT infrastructure to a socio–technical view where technology architecture, development processes, and IT skills are components (Bostrom et al. 1978; Henderson et al. 1994). In practice, IT skills and development (change) processes may prove to be an important factor. While the burden on the IT staff may tail off for a particular system implementation as time passes and the system operates routinely, the necessity of changing that system reintroduces the human influence to the infrastructure discussion. Previous research has examined in impact of IT skills on development (Byrd et al. 2001), but more recent research has elaborated on the human requirements for changing existing systems. With regard to changing systems, Shaft et al. parallels the assertions we are constructing here. That is, those responsible for making software changes must understand the existing system, the change itself, and their interaction to complete the solution (Shaft et al. 2006).

While we will be attentive to the human factors that influence the ability to change the applications as needed, we are primarily concerned with the technology characteristics of the application infrastructure that impact the ability of the IS department to change the applications as needed.

2.6.1 Technology Characteristics

In this study, we are trying to understand the impacts of the existing application infrastructure on the ability to change as needed in response to business needs. While changes to the broader IT architecture are certainly common, our assumption is that the hardware and

networking aspects of the infrastructure are relatively stable and independent with regard to changes to the application infrastructure. We are examining business-driven IS changes which typically occur at the application and data levels.

For years, researchers have been trying to label the aspects of software that enable it to be changed more or less easily. Research has indicated a number of reasons why infrastructure change is difficult or constrained including: incompatible systems, inconsistent or localized data models, old system architectures (Broadbent et al. 1999), hard-coded business processes, and links between applications (van Oosterhout et al. 2006). Additionally, research has suggested that complexity and development practices can have an effect on the maintenance of software (Banker et al. 1998). Among those factors that continue to appear as key structural components affecting the ability to change are integration and modularity (Byrd et al. 2001; Duncan 1995). Integration and modularity will serve as starting points to creating a list of application characteristics relevant to change.

Integration can be simply defined as the linking of information between various systems. At the data level, integration has been defined as "the use of common field definitions and codes across different parts of the organization" (Goodhue et al. 1992). At the process level, integration can be defined as linking the steps necessary to conduct business processes on separate systems. In application infrastructures, the connections between systems due to integration can be a barrier to changes in the applications.

A modular design is one that breaks up "a complex system into discrete pieces" to reduce an "unmanageable spaghetti tangle of systemic interconnections" (Langlois 2002). An alternate definition is: "A module is a unit whose structural elements are powerfully connected among themselves and relatively weakly connected to elements in other units" (Baldwin et al. 2000). In

software, modularity can be found at various levels depending on design. In general, software can be modular at the code-level with classes and objects, at the application level, at the add-on or plug-in level, and at the service level. At this point, there is no clear delineation between modularity levels, and quite often, the application infrastructure in organizations is a combination of systems with different levels of modularity.

For this research, we are interested in those aspects of the application infrastructure that affect the ability to make the required IS changes as opposed to creating a full description of the infrastructure. In some cases, the level of integration may cause ripple effects in other systems inhibiting the change. In others cases, the structure of the applications might allow changes to be easily performed.

2.7 INFORMATION SYSTEMS RESPONSE

In this study, we are examining the impacts of business-driven IS change requests that enter the application infrastructure domain, and the ability of the applications to respond as needed. In turn, the particular response can have effects on the performance of the business by constraining or enabling change in the business processes (see Figure 2-2). However, given an existing application infrastructure is in place, IS departments must consider the impacts of the IS change in conjunction with the existing application infrastructure. Together, the particular IS change needed and the particular application characteristics in place combine to limit or enable IS responses (Figure 2-4).

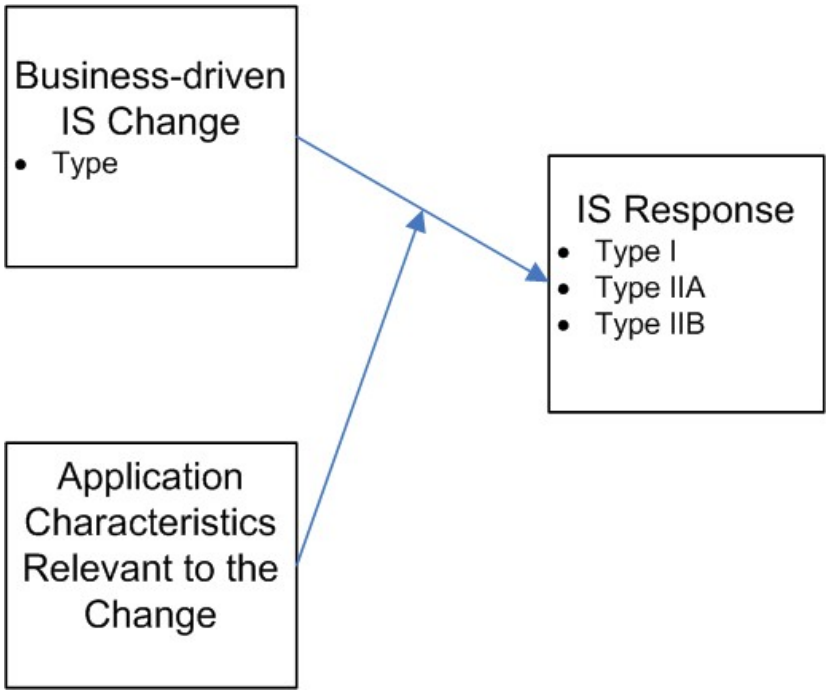

Figure 2-4: IS Response Model

This research distinguishes between three different types of IS responses based on three concepts: flexibility, agility, and expansion. Two important distinctions of this research should be made here. First, this research uses flexibility and agility research as a base for conceptualizing the ability to change. However, as will be described later, this literature is currently plagued with confusion due to overlapping definitions and inconsistent use. Second, this research does not view flexibility or agility as properties of a system, but as potential responses based on the interaction of the particular IS change and the existing application infrastructure. In other words, flexibility and agility are context-specific. These two distinctions together lead us to develop new terminology. Consequently, we term the three IS responses as Type I, Type IIA, and Type IIB.

2.7.1 Flexibility

Flexibility research has a long history in management, economics, and operations research, though it has been defined differently at times. Flexibility has been examined at the strategic level (e.g. (Eppink 1978; Grewal et al. 2001; Sanchez 1997)), the operational level (de Groote 1994), and even down to the plant equipment level (Browne et al. 1984). Following is a table showing selected definitions from various literature.

Table 2-1: Selected Definitions of Flexibility

Definition	Context	Source
"Strategic flexibility represents the organizational ability to manage economic and political risks by promptly responding in a proactive or reactive manner to market threats and opportunities"	Organization Strategy	(Grewal et al. 2001)
"the ability to respond rapidly and effectively to emergent needs or opportunities"	IT Infrastructure	(Duncan 1995)
"a characteristic of an organization that makes it less vulnerable for or puts it in a better position to respond successfully to unforeseen environmental change"	Organization Strategy	(Eppink 1978)
Type I: "refers to the firm's positioning of itself in such a way that it can deal with the occurrence of foreseeable events" Type II: "concerned with the ability to make good use of newly disclosed opportunities"	Economics	(Carlsson 1989; Klein 1984)
"the capacity to adapt"	Information Systems	(Golden et al. 2000)
"an organization's various abilities to respond effectively to various aspects of a changing competitive environment."	Organization Strategy	(Sanchez 1997)
a technology is "more flexible than another if an increase in the diversity of the environment yields a more desirable change in performance than the change that [one] would obtain with the other technology under the same conditions"	Operations	(de Groote 1994)

From the definitions, we can see that the various definitions indicate that flexibility is 1) a response to 2) the changing environment. In addition, Klein (1984) and Eppink (1978) have indicated another dimension: uncertainty. Eppink's work specifically mentions flexibility as protection against *unforeseen* environmental change, while Klein's later work describes a difference between environmental change that is *foreseeable* and the prospect of unforeseen

possibilities. Eppink sees flexibility as a tool for uncertainty management while Klein sees (Type I) flexibility as risk management (Carlsson 1989). The distinction between uncertainty management and risk management is important because of their different operating assumptions. Uncertainty management is based on managing the unpredictable, while risk management is concerned with managing among options of varying likelihood.

In the next section, we note that research on agility has begun to pick up on the same distinction between responding to foreseeable and unforeseeable change. This research will use this distinction because of the particular importance of these two types of change within the realm of IS.

2.7.2 Agility

The recent rise in popularity of the term "agility" has resulted in confusion about the definition of this term. In the following table, there seems to be considerable overlap in the agility and flexibility definitions. In addition to the overlap, some authors describe flexibility as a component of agility (e.g. (Conboy et al. 2004; van Oosterhout et al. 2006)) while others view agility as a component of flexibility (e.g. (Patten et al. 2005)). It is no wonder that measurement of these concepts has been so difficult.

Table 2-2: Selected Definitions of Agility

Definition	Context	Source
"the ability to detect opportunities and seize those competitive market opportunities by assembling the requisite assets, knowledge, and relationships with speed and surprise"	Information Systems	(Sambamurthy et al. 2003)
"responding to changes (anticipated or unexpected) in proper ways and due time" and "exploiting changes and taking advantage of changes as opportunities"	Manufacturing	(Sharifi et al. 2001)
"to be able to deal with unpredictable changes in market or customer demands"	Innovation Management	(Wadhwa et al. 2003)
"the ability of an enterprise to develop and exploit its inter- and intra-organizational capabilities to successfully compete in an uncertain and unpredictable business environment"	Manufacturing	(Hooper et al. 2001)
"the ability of an organization to thrive in a continuously changing, unpredictable business environment"	Knowledge Management	(Dove 1999)
"the ability to sense environmental change and respond readily"	Organization	(Overby et al. 2006)
"being able to act quickly on both the strategic and operational level to such unpredictable changes requires a new level of flexibility, which we call agility"	Information Systems	(van Oosterhout et al. 2006)

Fortunately, some recent research has begun to distinguish between flexibility and agility in a way that may allow their application and measurement to be separated. While agility and flexibility are both 1) responses to 2) environmental change, this research will leverage the categorization of Type I and Type II flexibility (Klein 1984). Specifically, the aforementioned separation of Type I and Type II flexibility distinguishes between the characteristics of the environment being responded to. Type I flexibility is a response to a known set of possibilities or parameters, which allows for planning and risk management. Type II flexibility is a response to newly disclosed opportunities or unforeseen events and is very similar to the definition of agility (Wadhwa et al. 2002).

2.7.3 Comparing Flexibility and Agility

In everyday use, we might describe two different definitions of flexibility and agility. Flexibility can be described in the context of stretching, where one's flexibility is related to how

close one comes to their physical limits. Agility is typically described as a response to being knocked off balance, much like a cat regains its balance (Dove 1999).

In more academic terms, de Groote has identified environmental diversity as the idea of "variability, variety, or complexity" (de Groote 1994). In his view, flexibility is a hedge against environmental diversity, with the implication that a "more flexible technology" can be chosen to offset the diversity. In other words, if one knows the environmental diversity which will be encountered, one can invest in the option that gives flexibility for that particular diversity.

However, this is not always possible. Quite often, the environment requires changes that are not within the parameters or the expected range of the original design. In other words, the variability, variety, or complexity of the environment was greater than the original design anticipated. This level of diversity requires more than reconfiguration changes to the technology. It requires redesign or redevelopment at a more fundamental level.

An examination of the tables above indicates many commonalities between the definitions given in research for flexibility and agility. Given the overlap in definitions and confusion surrounding the use of the terms flexibility and agility, this research will move to a categorization scheme derived from Klein (Klein 1984). Using the distinction between foreseen and unforeseen environmental changes as the boundary between Type I and Type II flexibility, we define two response types in the context of changing information systems as:

Type I Response – a response to *foreseen* diversity from the IS environment

Type II Response – a response to *unforeseen* diversity from the IS environment

2.7.4 Type I Response

To reiterate, a Type I response is a response to foreseen diversity from the IS environment. The implication of this response type is that the possibility of a particular IS

change was anticipated, and consequently the ability to make that change not only could have been, but was designed into the system in advance. Once the anticipated change is encountered, it is simply a reconfiguration change for the IS personnel.

The implication of anticipating particular change, or diversity, possibilities is that a Type I response (often through reconfiguration) is in effect a response prior to the occurrence of the change need. That is, a range of environmental diversity possibilities are planned for during the design phase. Consequently, additional time and effort must be devoted during the design phase to consider the change possibilities for the system, and the likelihood those changes will ever be necessary. This extra time and energy in development adds costs to the development process, which ideally are reclaimed through easier changes in the future.

Examples of Type I responses may be changing a tax table, adding a new vendor, or countless other possibilities. Regardless of the particular situation, the point is that for an organization to take advantage of rapid, reconfiguration type changes, the possibility for that change must be anticipated, whether the developer is internal or external.

2.7.5 Type II Responses

A Type II Response is a response to unforeseen diversity from the IS environment. In other words, the particular IS change was not anticipated (or if it was, it was not designed into the system) and therefore, the system cannot simply be reconfigured. Consequently, a Type II Response is always a reactive response in this context, and by definition, fits outside of the parameters of the original system design.

There are two subcategories of Type II Responses: Type IIA and Type IIB. A Type IIA Response is a response to unforeseen diversity, but the solution is within the scope of the system.

In other words, when reconfiguration is not an option, IS personnel can modify the system[6] at a more fundamental level, often through recoding, to meet the IS change needs. This response option is consistent with part of the IS maintenance literature where maintenance is initiated by a business change (Banker et al. 1998).

A Type IIB Response is also a response to unforeseen diversity, but the solution is outside the scope of the original system. Therefore, many organizations find that adding a system to the infrastructure is a simpler, quicker, and more cost effective solution than recoding. Additionally, many vendors, particularly ERP vendors, will not support modified systems making many organizations wary of changes to the ERP. Often, organizations find it more advantageous to add a system and integrate it with the ERP.

2.7.6 Assessing the IS Responses

There are several important points to emphasize about these responses. First, the IS response is not necessarily a choice. The IS response is often determined well before the change is encountered. It simply manifests itself when the change is needed. In other words, during the design phase of the software (whether it is internally or externally designed), and during the selection and implementation phases, particular environmental diversity possibilities are being prepared for. If the needed change falls within the parameters anticipated and designed for, then the IT department will respond with a Type I response[7]. If the needed change falls outside of the anticipated and implemented possibilities, either the system will be modified at a more fundamental level (Type IIA response), or new systems will be added to the infrastructure (Type

[6] At times, this will be a request to a vendor or development partner for a coding change.
[7] This assumes the IT staff has the skill to recognize these options before embarking on more substantial solutions. If the staff lacks the ability to recognize this possibility, then additional effort will be expended to "reinvent the wheel".

IIB response) (Figure 2-5). To clarify, if a Type I response is not an option, there is often a choice between Type IIA and Type IIB responses.

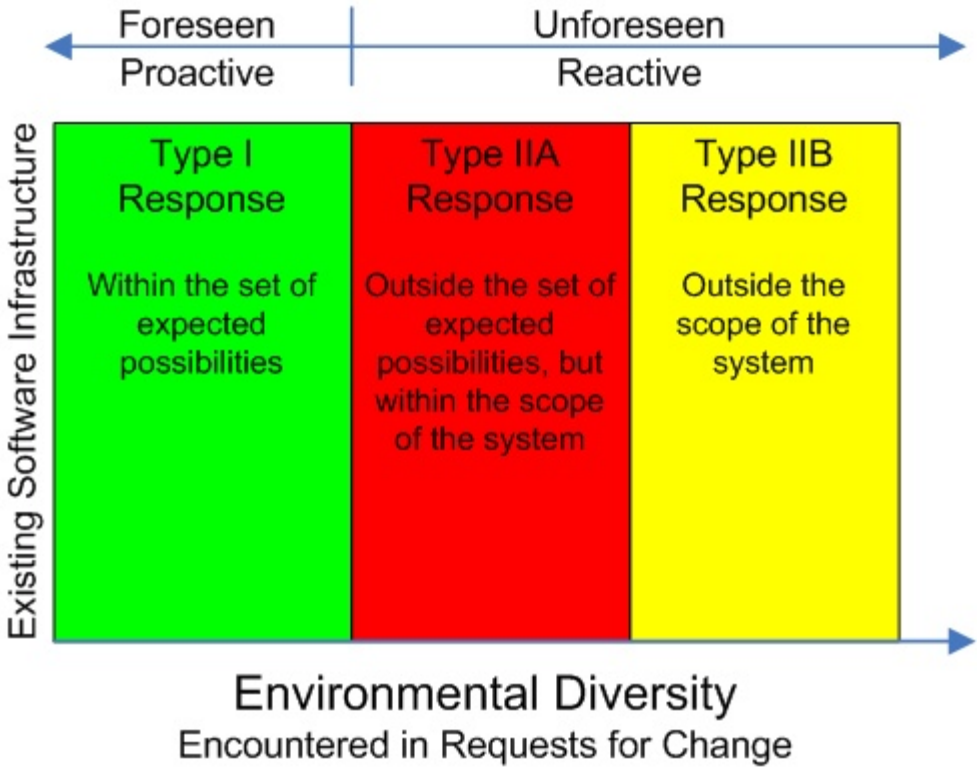

Figure 2-5: IS Responses for Varying Degrees of Environmental Diversity

Second, while researchers have characterized flexibility as a "property" of a system (de Groote 1994), the true flexibility of the system is context-specific. It is necessary to understand the change as well before flexibility can be assessed. This is a departure from many views of flexibility. Consider the following example of a technology with function A. Current setup (or configuration 1) may suffice for operation of the technology until change is demanded from the technology's environment. A technology "flexible" in function A may have additional setup options 2, 3 or 4. However, the technology can only be considered flexible if setup 2, 3, or 4

sufficiently meets the change need. If these options are insufficient, the technology must be considered insufficiently flexible in function A, and alternate options must be generated.

Third, the IS responses are expected to vary in several dimensions. Specifically, we expect these responses to vary in cost, difficulty, and time (Table 2-3). At the time of the change, having the option of a Type I response should be the least expensive, difficult, and time consuming. However, to be prepared to respond in a Type I manner, time and money must be devoted to predicting what changes may be necessary and incorporating the appropriate reconfiguration options into the system well before any of them are needed. Additionally, there is the concern that the time and money invested may be futile if the necessary changes fall outside of those anticipated at design time[8].

Alternatively, if the organization decides to devote less time to building or implementing a flexible software infrastructure, then they can expect to have the more expensive, time consuming, and difficult options of a Type IIA or Type IIB response.

Table 2-3: Criteria for Comparing IS Responses

Criterion	Description
Upfront Costs	Cost of preparing applications for eventual IS changes
Upfront Speed	Speed of preparing applications for eventual IS changes
Upfront Difficulty	Difficulty of preparing applications for eventual IS changes
Response Costs	Cost of accommodating IS change at the time of the change
Response Speed	Speed to accommodate the IS change at the time of the change
Response Difficulty	Difficulty of accommodating the IS change at the time of the change

Fourth, it is not necessary for the organizational response to match the IS response. In other words, an agile response for the organization does not necessarily require a Type II response at the IS level. Two scenarios illustrate this possibility. First, consider the need for an agile response at the organizational level for a manufacturing facility where part of the solution

[8] In the case of software purchase, the vendor is often paid a premium for more flexible systems.

is to begin manufacturing a part in-house rather than purchasing the part from a supplier. While this may translate to an agile maneuver for the organization in the marketplace, perhaps by undercutting a competitor's cost, it is likely that the organization's IS can simply be reconfigured (a Type I response) to change the supplier from external to internal.

Alternatively, the organization may decide that they need to change a particular business process. At the organizational level, this change may involve adding a step to the process, such as adding a supervisor's approval. In concept, this is a simple change for the organization, but at times, the IS department will find that this change requires recoding part of the system (a Type IIA response), rather than simply reconfiguring.

2.8 SUMMARY OF CONCEPTUAL DEVELOPMENT

In this chapter, we have attempted to accomplish several things. First, we discussed the prior research on agility, both outside and within IS, to identify the particular context of this study. In particular, this study concentrates on the ability of the application infrastructure to change in response to business needs, thereby enabling (or hindering) agility at the organizational level.

We also developed a model of change to application infrastructure that diverges somewhat from previous conceptualizations. Rather than considering flexibility or agility a property of the system, we propose that these comprise the IS responses (along with expansion) that cannot be evaluated until the particular change and the particular application are considered. Put another way, the IS department cannot know if they will respond in a Type I manner (reconfiguration), or if they must choose between a Type IIA response (recoding) or a Type IIB response (system expansion), until they know what change is required to the particular system.

The three types of IS responses are derived from previous research on flexibility and agility, as well as preliminary interviews with practitioners (which will be discussed later).

This reconceptualization of the interaction between IS changes and the IS infrastructure helps us construct a model (Figure 2-6) to guide this research. This model indicates the three major areas of interest for this study which correspond to the research questions. First, we create a typology of high-level business-driven IS changes that are encountered in organizations (RQ1: What are the types of change required of the IS by the business?). Additionally, we will gather information on whether particular business-driven IS changes were foreseen prior to their occurrence. Second, we examine the application characteristics that are relevant to IS changes to get a better idea of the factors that enable or inhibit change (RQ2: What are the application characteristics that impact the ability to change?). Finally, we examine the IS responses along and their occurrence in particular change situations (RQ3: What are the response options of the IS, given a particular IS change need and a particular software infrastructure?).

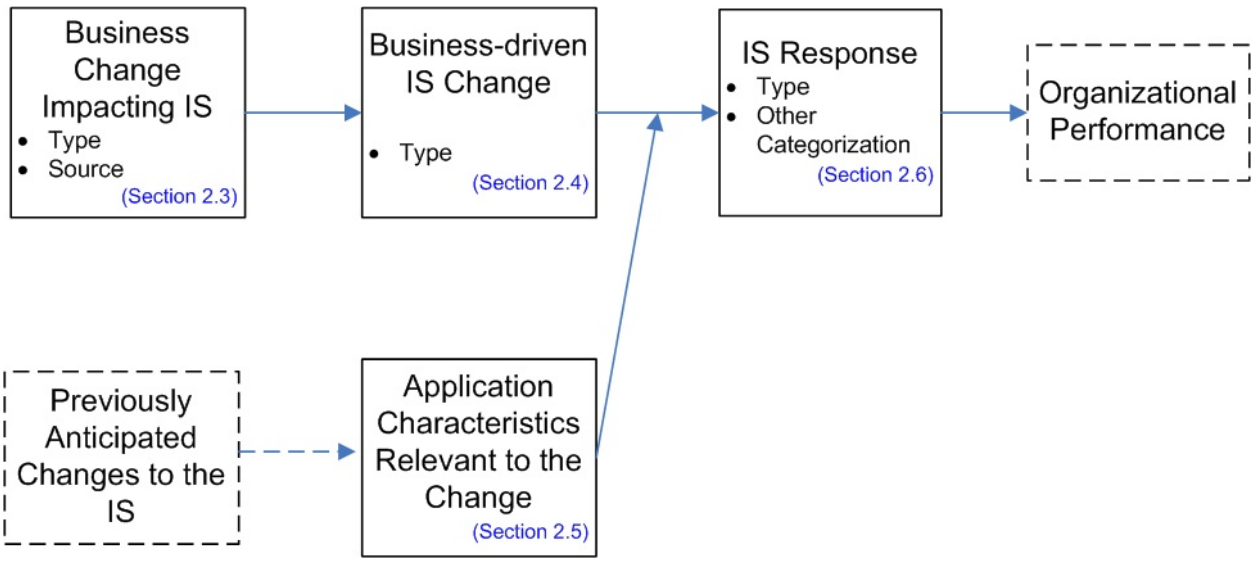

Figure 2-6: Initial Conceptual Model (Revisited)

This model serves as the framework and guide for conducting this research. Although much of this research is exploratory, development of this model allows the data collection and analysis process to be more structured and focused. In the following chapter, we will discuss the method that will be used to collect data for this research.

CHAPTER 3 RESEARCH METHOD

Perhaps as a consequence of its relative youth in the IS field, research on the ability to change IS in response to business needs is underdeveloped. The relevant literature base for this research is IS flexibility and agility research (and its related concepts) given their connection to change, but there are few theoretical concepts that are defined well enough to build quantitative measurement schemes for empirical study. Additionally, given the multi-faceted, multi-level conceptualizations of flexibility and agility described in previous chapters, we must be more careful and thorough in defining concepts as we move forward in this research area.

To that end, this chapter describes the qualitative methods that were used to better understand the phenomena surrounding changing information systems in response to business needs. This chapter discusses the reasons behind method selection as well as site selection, interviews conducted, the preliminary coding and final coding schemes, and the analysis techniques.

3.1 PHILOSOPHICAL FOUNDATIONS

Given the previous lack of focus in defining agility and flexibility in academic research, we have chosen to study the research questions using semi-structured interviews. It is important to be specific about the ontological perspective taken when using qualitative techniques as qualitative studies can be conducted from positivist, interpretivist, critical, or realist perspectives (Ackroyd et al. 2000).

Given that this research is being guided by a prescriptive model which has yet to be verified in the field, we must be careful in examining what we expect to find during data

collection. That is, we may find little direct evidence of the existence of this model in the minds of practitioners. However, this does not necessarily discredit this model as an underlying "force" dictating behavior. We may, in fact, find evidence of a different model.

This research adopts a positivist perspective. Ontologically, positivists believe that a reality exists independent of our ability to observe it (Orlikowski et al. 1991). In general, we expect an underlying reality to exist and it is our goal to observe it. Therefore, epistemologically, we have constructed an a priori model leaning on logic and previous research, and expect that interviews will allow us to discover this reality. To the degree that we are successful in observing the reality, we expect three or more possibilities to occur during data collection for this study.

> 1: Practitioners perceive a similar model of the change – application infrastructure interaction, as well as the responses.
>
> 2: Practitioners perceive a different model of the impacts of change on the IS and how it leads them to particular responses.
>
> 3: Practitioners do not have a well-formed model within the realm of changing IS.

Each of these possibilities provides interesting outcomes for this research. In case 1, if practitioners perceive a model similar to the prescriptive model in this research, then the model can be considered qualitatively substantiated and further development on quantitative measurement can proceed. If case 2 emerges, we can contrast the differences between the models to identify gaps and attempt to reconcile them. If case 3 is evident, then the prescriptive model may prove highly useful for practitioners in organizing efforts to manage change to their application infrastructure.

3.2 UNIT OF ANALYSIS

The development of the ideas contained in this research began with the general notion that the application infrastructure has characteristics that inhibit or enable change to the applications comprising the infrastructure. Since we are particularly interested in business-driven IS changes, we have focused on the key applications and data due to their connection to the execution of the business processes.

However, for detailed study and data collection purposes, consideration of the entire application infrastructure of a large organization is impractical for a couple of reasons. First, large organizations can have hundreds of applications that form their infrastructures. Including all of them would dilute our ability to make inferences about what we consider the most important systems – the key applications. Second, individuals in organizations do not typically have detailed knowledge of all of the systems and the changes that those systems undergo. They usually have knowledge of a subset of the systems at a level of detail that is useful for this research.

Therefore, we developed an interview template that targeted *changes to particular software systems* that participants are familiar with. Our goal in targeting changes to particular systems as our unit of analysis was to get specific examples of changes, both business and IS, the characteristics of the system that are relevant to the change, and the particular responses that are available for handling the change.

Upon evaluation of the interview data, it became apparent that analysis of changes to particular software systems would be difficult primarily due to the lack of consistency between change examples. Most notably, informants frequently skipped back and forth from real examples to hypothetical situations. In some examples, information was lacking on one of more

of the primary constructs, which made it difficult to confidently link application characteristics relevant to change with particular business-driven IS changes or particular IS responses.

As a consequence, it was decided to elevate the unit of analysis first to the interview level, and second to the organization level, which are both utilized to report the results in subsequent sections. This allowed us to see evidence of the high level constructs in the initial conceptual model (business-driven IS change, application characteristics relevant to change, and IS responses) as well as subcategories (e.g. modular designs, reconfigurable designs, dormant functionality, user exits, etc. under the application characteristics relevant to change construct). However, this elevation of the unit of analysis eliminated the possibility of clear examination of the interaction effect between the business-driven IS change and the application characteristics relevant to change. Future research could attempt to be more consistent in keeping emphasis on real examples of IS change and the associated constructs. However, this level of analysis did allow us to better understand (through statements about specific and hypothetical examples) how these individuals thought about the issue of business-driven change, the application characteristics, and the possibilities of change.

3.3 METHOD SELECTION

This research used semi-structured interviews with industry IT practitioners as the primary data collection method. There were three primary reasons for the selection of semi-structured interviews as the method for this research. First, as previously mentioned, academic research devoted to understanding the issues around changing information systems in response to business needs is relatively young. Consequently, the research questions for this study do not lend themselves to using established quantitative measures at this time. Interviews will allow for

deeper explanation and exploration of the concepts and issues surrounding a youthful research area.

Second, although the precise constructs that are important are not clear, a prescriptive model was developed using logic to serve as a guide for structuring this research (Miles et al. 1984). It was unclear what aspects of the model are perceived by practitioners, to what degree it guides their decision making and performance, or whether the model contains the most relevant constructs. Hence, this research sought to establish to what extent the prescriptive model fits with practitioners' perceptions. Semi-structured interviews allowed flexibility to discover whether the proposed model or other underlying models exist in the minds of practitioners.

Third, interviews allow for opportunistic discussions to arise. Since it was expected that different individuals would have different perspectives about IS change, the interview method allowed for more exploration to uncover details surrounding the concepts of this study that might have been unseen.

3.4 SITE SELECTION AND INTERVIEW SUMMARY

Interviews were conducted at five large organizations in different industries. There were a few restrictions on the companies selected. First, the company had to be large enough to have an IT department. Second, the IT departments had to participate in at least some of their own software development. These five organizations were a part of a convenience sample but are considered typical large organizations. The organizations' pseudonyms indicate the industry they represent.

Within each organization, interviews were conducted with as many individuals within the IT department as allowed. Table 3-1 indicates the organizations and informants involved in this

study. The third column indicates whether the interview was included in the data analysis (coding) for this study.

Table 3-1: Organizations and Informants Included in This Study

Organization	Informants	Included in Analysis
PowerCo		
	1. CIO	Yes
	2. Sr. Program Manager	Yes
	3. Business Analyst A	Yes
	4. Business Analyst B	Yes
	5. IT Architect	Yes
InsuranceCo		
	6. CIO	No
	7. VP System Implementation	Yes
	8. VP IT Planning w/ Jr. IT Planner	No
	9. Sr. IS Manager	Yes
	10. IT Development Manager	Yes
	11. Sr. Developer A	Yes
	12. Sr. Developer B	Yes
SoftwareCo		
	13. CIO	Yes
	14. Director Corporate Application Support	Yes
	15. Business Analyst A	Yes
	16. Business Analyst B	Yes
	17. Business Analyst C	Yes
RetailCo		
	18. CIO	No
	19. CTO	No
	20. Director of Business Process & Technology	Yes
	21. Manager of IT Strategic Planning	Yes
	22. Sr. IS Manager, Sales	Yes
	23. Sr. Developer A, B, and C	Yes
	24. ETL Manager	Yes
BankCo		
	26. Sr. Director of IT	No
	27. SOA Manager	Yes

In total, twenty-seven field interview sessions were conducted with thirty-one individuals from five organizations (a few sessions had multiple informants present). Of these, twenty-one interview sessions were retained for data analysis. While field notes gathered from the excluded

six interviews were used to inform the research, they were dropped from the coding process primarily because the interviews were not recorded at the request of the informant. The informants included upper level managers, IT department managers, business analysts, and developers.

The interviews were semi-structured using a pre-designed script (see appendices A and B) of open-ended questions based on the major constructs and relationships we wanted to examine. Follow-up questions were used where pertinent to gather more detail about a particular example or further investigate interesting viewpoints. By design, the interviews began with open-ended questions, but gradually got more targeted as the interview progressed. For example, if a respondent did not mention expansion (Type IIB response) in a IS change example, they were more directly asked about expansion by the end of the interview. This allowed the participant to discuss change to an application of their choice with as little interviewer bias as possible. Interviews ranged from about 25 minutes to 1.5 hours, with an average of approximately 1 hour. Interviews with participants were face-to-face in all but two sessions.

The questions in the interview scripts were tweaked based on results of a series of five pilot interviews within PowerCo. The pilot study allowed for questions to be revised in order to be more effective in examining the issues of interest for this study. While changes were made to the interview scripts, they were incremental, and therefore, the data gathered from PowerCo was folded into the final analysis for this work.

The interview script was partially adapted for the role of the individual being interviewed since the knowledge level and knowledge base of the individuals varies. For example, the CIO will have more insights about the business changes driving change efforts within his department,

while the developer will have better understanding of the characteristics of the applications that allow or inhibit change.

Informants were allowed to discuss experiences and systems that they had personally dealt with in their job. This did lead to some overlap in systems discussed, but overlap of systems was not a requirement. This decision was made so that examples would be more concrete. An alternate design might have required discussions to center around a single system within an organization, but it was considered preferable to use examples that informants were most comfortable with.

When possible, the interview was recorded for transcription purposes. Recording of the interviews reduces "variation in observation" when creating codes and updating the coding template during data analysis iterations (Boyatzis 1998).

By conducting interviews within several different organizations with individuals in different roles, this study has benefited by casting a wide net. The broad range of organizations in various industries allowed this study to discover whether business-driven IS change issues were present in a variety of companies. By speaking with a range of informants in different roles, this study was able to collect a number of perspectives around business-driven IS change needs. Each of these perspectives were potentially different because of their varying "distance" from actual IS change at the code level.

A couple of alternate designs for this study might have yielded different results. For instance, a single case study approach might have been used. Within a single organization, an exhaustive set of informants could have been interviewed using a similar interview template. This might have yielded more detail about that organization and their issues with business-driven IS change, but it would have sacrificed some generalizability to other organizations.

A second approach would have been to interview individuals in the same role across several organizations. For example, developers might have been targeted as individuals dealing with business-driven IS change. This approach might have yielded a deeper understanding of that role with regard to the topics of this study, but again, it would have sacrificed some generalizability to IT personnel as a group.

Acknowledging that no study is perfect and all studies must make tradeoffs in execution, this study settled on a broad approach for preliminary study of the issues surrounding business-driven IS change. Further research will have opportunities to dig deeper using the findings in this study as a starting point.

3.5 DATA ANALYSIS

Data analysis began shortly after data collection started. Since this is an exploratory study, it was important to continually assess and improve the interview template to maximize the content of the interviews. After conducting the interviews in the first organization, an important new finding related to anticipation of IS change emerged. While it was included in the initial conceptualization, it was initially not a central focus of the research. In addition to the initial interview plan, subsequent interviews included more investigation into this new area.

Interview coding started following the interviews within the third organization. This research employed a hybrid coding technique, utilizing both theory-driven code development and inductively-derived codes (Boyatzis 1998). This allowed the research to use previous research for guidance while considering new ideas and themes that emerged from the interview data. Code development began with an initial template (Appendix C) based on the model developed in Chapter 2, but evolved as interviews were transcribed and analyzed.

Two coders were utilized to validate the coding scheme and determine the interrater reliability of the coding scheme. Interrater reliability allows the researcher to evaluate the consistency of matching codes to informant responses (Boyatzis 1998; Straub et al. 2004). The final coding scheme (Appendix D) was iteratively developed based on discussion following coding sessions. The second coder studied ten of the final twenty-one interviews and coded seven. The following process was used on each interview in turn. First, each interview was coded independently by both researchers using the latest coding template revision. The codes were then compared and discussed. Where there was disagreement on a code or a new code was necessary, the code was discussed until a consensus was reached and the coding template adjusted by adding a new code or refining the definition of an existing code. Then the process continued with the next interview. After the coding scheme was stabilized, the primary researcher used the final coding scheme to recode the interviews used to tweak the coding scheme.

Interrater reliability was calculated for each multi-coded interview based on the calculation: (# of code matches / (# of code matches + # of code mismatches)). Interviews were coded until interrater reliability reached an acceptable level. While some researchers suggest that 0.70 is an acceptable interrater reliability coefficient (Miles et al. 1984), this study continued with the coding process until at least 0.80 was reached. While several interviews exceeded the 0.70 threshold, the coders considered their coding efforts to be consistent upon reaching an interrater reliability of 0.845. This level of agreement was reached at the construct level. More refinement will be needed for sub-code levels in subsequent studies.

In qualitative studies, researchers strive for theoretical saturation, where “no new or relevant data seems to emerge” (Boudreau 2002). At the major construct level for this research

(highest level codes), we believe that this has been achieved. Subsequent research will attempt to achieve theoretical saturation at the more detailed levels of the model (subcodes).

Each interview resulted in a profile of codes, or a listing of whether the code was found in the interview or not. Since the results are reported at the interview level rather than the change level, multiple instances within an interview of the same code were considered redundant. In other words, an informant mentioning reconfigurable designs on multiple occasions was tabulated as one. Within each company, the number of interviews where a code was mentioned is listed in the tables of the following chapter.

CHAPTER 4 RESEARCH FINDINGS

4.1 STRUCTURE OF THIS CHAPTER

This research seeks to describe and understand the types of IS change driven by business changes, the impact of existing applications on those IS changes, and the response options of the organization given a need for IS change and a particular application infrastructure. The previously developed ideas are predicated on the principle that change to the IS is not independent of the existing application infrastructure. In fact, given the specific and defined nature of IS internally, the ability to change rapidly and with ease as needed is predetermined by previous decisions about the applications that make it more or less easy to meet a specific request for change.

This chapter begins with a discussion of the findings related to the initial conceptual model's constructs along with illustrative quotations from the informants. New findings are then discussed along with a new model derived from the interviews.

4.2 RESPONDING TO IS CHANGE NEEDS

Prior to completion of the semi-structured interviews for this research, this research was guided by the initial conceptual model (Figure 2-3, reintroduced in Figure 4-1). In essence, the initial conceptual model argues that the ability of IS to make changes quickly and easily cannot be predicted without consideration of application characteristics relevant to that change. Another way of saying this is that the IS response is dependant on, if not dictated by, both the particular IS change required and the application characteristics relevant to that change.

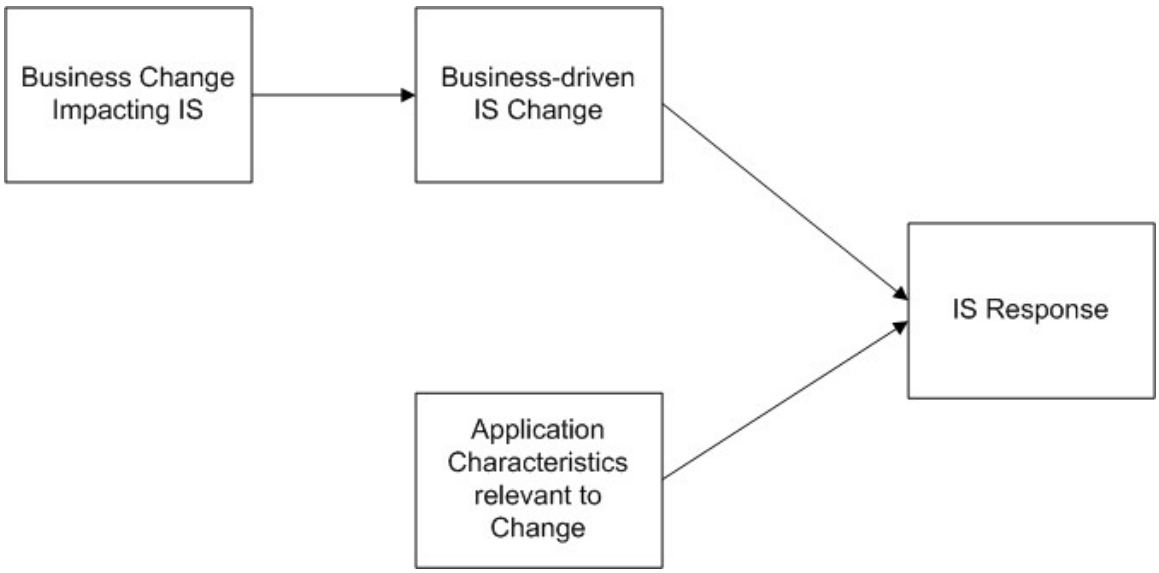

Figure 4-1: IS Response – Initial Conceptual Model (Revisited)

In one example[9] of business-driven IS change, the CIO of SoftCo discusses the business need to synchronize data in multiple systems. This business-driven IS change was based on a business need to have all parties of the organization reporting the same prices for products. The lack of synchronization meant that there was potential for salespeople to have different prices than the service organization or the website. The old process was to devote an individual to the task of maintaining price synchronization across the various applications. This business need led to IS changes related to data integration and new processes. Furthermore, it can be seen that these particular IS changes are what we will call interdependency changes, or changes related to the interfaces of applications. This is one of several categories of IS change that emerge from the interviews.

[9] This example is also related to the new findings and will be discussed accordingly in following sections.

> "So there's an SKU code for every product we sell on a price. And there's a set of tables that describes the service, the product pricing, the usage, the descriptions, and so forth that people can purchase. And we had a scenario where we had multiple systems where that data needed to be present. And we have a sales system that is separated from our financial system. Now on our web site, you know, we need to be able to show people product data. And then on our support system we need to know that when somebody calls in that they have purchased the appropriate product. So those same codes need to be immaculate. So what was happening was that there was an individual, well actually a couple of people, whose responsibility was to keep the pricing data in synch in all those different systems. So if we changed a price for a product, we had to go into each one of those systems and change it and hope, if you will, that they didn't miss anything … or that their business logic of getting from A to B to C was consistent."

Although, these applications did have user exits that would allow for data to be manipulated from outside the application, these user exits were inconsistent.

> "Factually speaking the systems by this definition aren't consistent because they were built by different vendors using different functions."

While the user exits were created to modify the needed fields, and were thus application characteristics relevant for these particular IS change needs, their inconsistency with one another did not allow SoftCo to use any one of the applications as the system of record or master file. If there had been more consistency between the systems in terms of user exits, SoftCo might have been able to recode some of the applications to work together. Since the Type IIA IS response, rewriting all or most of the separate systems, wasn't a practical solution in this case, SoftCo chose a Type IIB IS response of expansion. This expansion IS response resulted in a new application outside of the scope of the original applications whose task was to maintain data consistency through data integration and replace the current manual process of updating several systems.

> "…so what we did is we built a system, a custom application, that allows the users to create pricing in one system, a master record, and then pushes those changes to the other systems automatically. We call it Product SKU Integration. If I create a new piece of software and I add a price, I enter it in that application and it updates the financial system, the sales system, the customer relationship system, and training, database - whatever other environments are needed to be updated. I think five systems it actually touches and it keeps track in the back end of how all those translations work between systems A, B, and C. So in this system it might be called a price, and in this system it might be called a cost, here it might be called an inventory item or it might be called a SKU, whatever all those translations are there's this big secret table that keeps track of all that. The user doesn't need to worry about it. They just enter product and price and quantity, or whatever they do. The system does all that work behind the scenes in a consistent manner and we also have some clean up scripts to pick it all up from the beginning. So now you know that the product definition is consistent across throughout all of the products. That is really so important for reporting, for transactions. If I get a transaction on one system and look at it, well who sold it? Well I can go over here and see, here it was. We had a price that was the same. You don't have to worry about all those fundamental issues that we used to have."

This example illustrates all of the major constructs of the initial conceptual model – a set of business-driven IS changes (need for data consistency across systems), some application characteristics relevant to the change (not integrated, but with user exits), and a resulting IS response (expansion to add a new module talking to all systems). Furthermore, the example illustrates not only the existence of these constructs, but also the interaction between the particular business-driven IS changes and the application characteristics relevant to the change. In this case, the system had application characteristics which were needed for data integration, but they were insufficient to meet the need. Consequently, SoftCo was left with expansion as their best alternative for an IS response to the business need.

In the following sections, summary statistics of the interviews are discussed as they relate to the constructs of the initial conceptual model. Quotations from various informants will also be used to describe the constructs found in this model in richer detail. To give greater context to the quotes, other aspects of the model may be found in the quotes as well.

4.2.1 Business-driven IS Change

Business-driven IS change refers to those changes to applications that are required in response to business needs. This excludes those changes to applications based on technical needs only, such as errors in programming and technical upgrades. In this research, we are trying to understand the types of business-driven IS change required by the business (Figure 4-2).

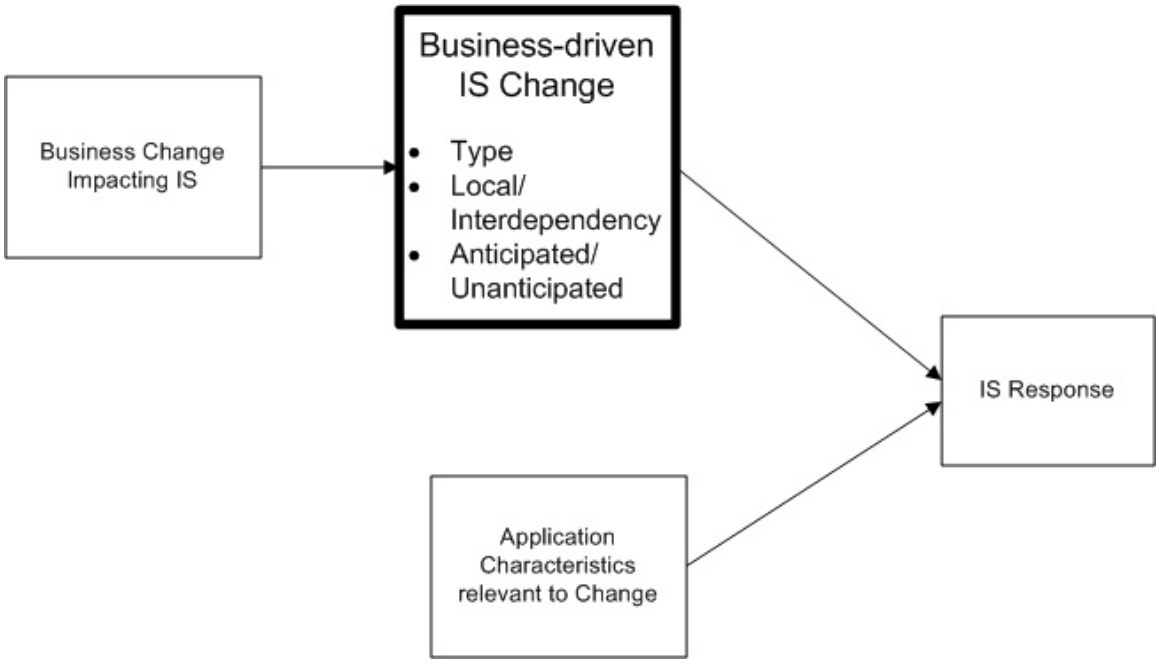

Figure 4-2: Business-driven IS Change Dimensions

Based on the interviews, we have examined the data on business-driven IS change from several different perspectives and have settled on three main categorizations. First, we searched for types of IS change required. The intent behind the categorization is to abstract up from the various IS changes encountered to a smaller, more parsimonious set of change types. This resulted in a list of general IS changes that does not imply a particular application or architecture. In other words, a reporting change is a type of IS change, but it does not imply that the reporting change happens to a legacy system, an ERP system, or otherwise. Second, IS changes were

categorized as either anticipated or unanticipated. Third, IS changes could be distinguished at times as either a local, or internal, change to a particular application, or an interdependency change where the relationship, or interface between applications were modified. Each of these perspectives on business-driven IS change illuminate different areas of concern for organizations.

4.2.1.1 Business-driven IS Change: Types of Change Required

The initial conceptual development of this research argued that business-driven IS change is related to organizational agility because efforts to be agile as an organization often require changes to the information systems of the organization. As a result, the concept of business-driven IS change became a central construct and the basis for one of the primary research questions in this research. From the beginning, it was hoped that a list of general types of IS change could be created where no particular application type was indicated. Fortunately, the interviews resulted in many different examples of business-driven IS change in varying levels of detail, where general types of IS change could be abstracted from the applications where they occurred. While the list is admittedly incomplete and open to debate, it serves as a starting point and basis for further development.

In the following table, seven different general types (Table 4-1) of business-driven IS change are depicted with their occurrence by company (Table 4-2). The number in each cell represents the number of informants within that company that mentions the particular type of IS change (i.e. in PowerCo there were five interviews. Four of five informants mentioned data integration changes, two of five mentioned user interface changes, and so on.) Where a cell in the table is blank, there was no mention of this type of change from anyone in the company. This should not necessarily be taken as evidence that this type of change does not occur. Instead,

it can only be safely stated that it was not mentioned. In other words, non-zero numbers in the table carry some information about the presence of these types of changes in the organization, but the lack of non-zero numbers may carry no information. Finally, the final column represents the percentage of interviews that mentioned a particular change type.

Table 4-1: Working Definitions of Types of IS Change

Type of IS Change	Definition
Data Integration	Connecting data components between applications
User Interface	A change to the presentation of information to the end user or a change in the interaction between the end user and the application
Communications	The implementation of new communications channels to meet a business need
Reporting	A change to the information presented in reports
Business Process Change	A change to an existing business process within an application or series of applications
New Functionality	Additional of functionality previously not include in the application
Data Collection	Addition of new data collection fields into applications or databases

Table 4-2: Business-driven IS Change - Types by Organization

Type of IS Change	RetailCo (of 5)	InsuranceCo (of 5)	SoftCo (of 5)	PowerCo (of 5)	BankCo (of 1)	Total (of 21)	Percentage (of 21)
Data Integration	1	1	5	4	1	12	57%
User Interface	2	1	3	2	1	9	43%
Communications	1				1	2	10%
Reporting	1	1	3	2		7	33%
Business Process	1		4	3	1	9	43%
New Functionality	2	2	1	3	1	9	43%
Data Collection		1	1	1		3	14%

Table 4-3 shows quotations that illustrate the various types of business-driven IS changes are shown.

Table 4-3: Business-driven IS Change - Illustrative Quotations of Types

General Type	Business-driven IS Change – Types	Company	Informant
Data Integration	"For example, our sales systems, we pull in a lot of data from maybe a third party that we would want to be displayed on a screen."	SoftCo	Business Analyst A
User Interface	"An out of the box application wouldn't show that we might, in whatever way the application allows, add those fields, text boxes to the front end."	SoftCo	Business Analyst A
Communications	"We are also using satellite phones, and we are trying to learn how to use GPSs even more for storms and outages."	PowerCo	IT Program Manager
Reporting	"Business needs recently…have been more requests for additional reporting…And now the users are wanting to get the information out and be able to do reports on them…They want additional reporting functionality and the data to do that."	InsuranceCo	Senior Developer A
Business Process Change	"How can we change the business process in order to grow share, and by that, that means application and effect…So anything that's related to a business process is open and possibly could change."	RetailCo	System Manager A
New Functionality	"…what tends to happen is that when we do new functionality built on top of what we've got, then we never go back and dismantle the old stuff."	RetailCo	System Manager C
Data Collection	"Some of the stuff is adding data, kind of tacking it along with something that might come out of a box."	SoftCo	Business Analyst A

4.2.1.2 Business-driven IS Change: Anticipated and Unanticipated

A second angle to view business-driven IS change from is whether the change was anticipated or not. While dependent on experience, IT personnel often have some expectations of what changes may be required from a business or technical perspective. If a change is anticipated, it may be more easily accommodated. Still, in many cases, they may not expect the necessary IS changes, and consequently, may be less certain initially how to accommodate the IS changes.

Table 4-4 shows the number of interviews per company where anticipated and unanticipated changes were mentioned. If at least one anticipated (unanticipated) change is mentioned in an interview, then that interview is counted as having that kind of change. In the interviews, when IS changes were discussed from a perspective indicating anticipation or failure

to anticipate, follow up questions would target examples of the opposite case. Consequently, it is not a surprise to see that the number of anticipated and unanticipated IS changes were about the same within each site. Interestingly, it does appear that more individuals at InsuranceCo frame their examples with discussion of expectations and being surprised by some of the IS changes requested. This may originate from any number of sources, both real and systemic. One consideration is that more developer-level individuals were interviewed at InsuranceCo relative to the other organizations. Relative to business analysts and higher-level management, developers may have a more concrete feel of which changes were foreseen and which blind-sided the organization.

Table 4-4: Interviews Indicating Anticipated and Unanticipated IS Changes by Organization

Type	RetailCo (of 5)	InsuranceCo (of 5)	SoftCo (of 5)	PowerCo (of 5)	BankCo (of 1)	Total (of 21)	Percentage (of 21)
Anticipated	1	4	2	2		9	43%
Unanticipated	3	4	2	2		11	52%

In some cases, informants stated explicitly that particular IS changes were anticipated (Table 4-5) or unanticipated (Table 4-6), while in others, informants implied whether the change was anticipated or not. In general, descriptions of anticipated changes indicated planning and specific steps taken to prepare for the anticipated change. Descriptions of unanticipated changes generally indicated, not necessarily a lack of awareness that change is possible, but that the particular change mentioned was unforeseen.

Table 4-5: Business-driven IS Change - Illustrative Quotations of Anticipated IS Changes

Business-driven IS Change – Anticipated	Company	Informant
"Eventually we'll want to get client data from other sources - from customers directly…let a customer come in and change his own address. So we anticipated that and the coding of that has all that in mind. All those hooks available for grabbing other data from other sources down the road."	InsuranceCo	VP of Systems Implementation
"There's what we call operational changes, which would be tax changes, small changes like that you don't really see, municipality changes, address changes. If you think of how many counties we have in the state, and each of those have something a little different." [The capability to change these easily by changing table values was built in.]	PowerCo	Business Analyst A

Table 4-6: Business-driven IS Change - Illustrative Quotations of Unanticipated Changes

Business-driven IS Change – Unanticipated	Company	Informant
"With regard to one of the things I mentioned earlier where we were setting a simple manual reason code, well that was one thing that…I never expected it to balloon out the way it did. I never expected the mainframe side of the house to come back and say well we also want to use it for this and we also want to use it for this, and we want to set it here, we want to set it there and we want to do that. And because that wasn't expected, it wasn't programmed in a modular way and that's what led to it being sent to a bunch of different, disparate places."	InsuranceCo	Senior Developer B
"A good example is we've been asked to integrate the claims tracking systems with some big client applications on a PC platform…our corporate push is to go thin, and I would never imagine we would be asked to do some integration with big client applications, but something opened a request for that."	InsuranceCo	IT Development Manager
"We have a client system where everything is attached to a central client database. In the past all client data was attached to individual policies which means it could differ from policy to policy, but we created this big huge database of all clients which would be customers, mortgagees, lawyers, glass companies, whoever, it's going to be entered in the system. You have to go to client system and pull that information. This particular PCR [Project Change Request] realized, hey there are some clients that touch thousands of customers, like mortgagees. And any of our service centers could go in there and change it. And if they did it's going to generate paper to everybody who's attached to that mortgagee. So we just didn't think about that in the beginning and this locks down certain clients and gives very limited access to make change to them. It makes a lot of sense. We've absolutely got to get that done."	InsuranceCo	VP of Systems Implementation

4.2.1.3 Business-driven IS Change: Local and Interdependency Changes

Another finding from analysis of the interviews is that the impact of business-driven IS changes was dependent on whether the change was local to particular applications or whether the

change was between systems (an interdependency change). This perspective emerged from the examples given by the informants and creates some interesting implications for software structure.

Local IS changes, while still potentially complex, only require changes that are "contained" within a particular application. There may even be cases where a business change requires local changes in several different applications. An alternate way to look at a local change is that the change does not affect its relationship with other systems. While there can be substantial complexity associated with local changes, as long as the inputs and outputs of the application do not change, there is conceptually no effect on its relationship with other systems.

Alternatively, interdependency changes are based on relationships between applications where information from one application is provided in a particular format and applications on the receiving end must be prepared to work with that format. Consequently, if the format of the information provided by one application changes, the applications receiving that information often require changes as well. As the number of systems that are interdependent increases, the potential for unforeseen and unintended consequences associated with interdependency changes increases drastically. In fact, interdependency changes typically require substantially more changes to verify that an intended interdependency change between two applications does not ripple through the relationships between the various other interdependent systems. As a result, IT personnel generally view local changes as easier to accommodate compared to interdependency changes.

A senior developer at InsuranceCo gave a brief description of the considerations when the IS changes are local or interdependency changes.

> "We look at the change that's being requested and we try to determine, what are the touch points? If I change this in this program, what other programs does this program affect? What night-time processes might this change in the data affect? Is there anything that feeds into this that needs to change?"

Table 4-7 shows that the local and interdependency IS change perspective did show up in a number of interviews. It is not unexpected that the numbers of good examples are low since this finding was emergent during analysis, and thus, not directly targeted in the interviews. It is assumed that the distinction between local and interdependency is rather obvious, but useful nonetheless considering the difficulty associated with each of these IS change types in different situations.

Table 4-7: Interviews Indicating Local and Interdependency IS Changes by Organization

Type	RetailCo (of 5)	InsuranceCo (of 5)	SoftCo (of 5)	PowerCo (of 5)	BankCo (of 1)	Total (of 21)	Percentage (of 21)
Local	1	3	1	1	1	7	33%
Interdependency	1	3	1		1	6	29%

According to informants, local IS changes were generally easier to accommodate because the difficulty of the change was limited to the application of interest (Table 4-8). As long as the change did not affect other applications by changing what is shared between applications, effort and testing was reduced relative to interdependency changes.

The testing aspect was an important issue to several informants. A senior developer at InsuranceCo described how local changes and interdependency changes affect the amount of testing effort required to make the changes.

> "It's easier to change that one little section and test it, without having to test everything. If I have to test everything, then that's a big effort because it's a big testing effort."

Table 4-8: Business-driven IS Change - Illustrative Quotations of Local IS Changes

Business-driven IS Change – Local	Company	Informant
"…anything that's in a single system would probably be the easiest to change. There's just not a lot of coding, not a lot of testing."	SoftCo	Business Analyst A
"Sometimes, if you're at the service level…you may say for instance, 'I want to record the sales event earlier in the process rather than later.' So if you reconstruct where those things occur in the process, that could cause you to reissue a service. If you don't change the contract and the inputs and outputs, you can do that without making any changes. And that's also the beauty of SOA – as long as you don't change the contract that you've issued, you can reorder things within the process with no interaction with the consumers of it."	BankCo	SOA Manager

Interdependency changes are often more complex because of the connections between applications (Table 4-9). Informants indicated that there were concerns about rippling effects to additional systems when changes are made, which often leads to much more evaluation and testing of changes before deployment, slowing the effort to change the applications as needed.

Table 4-9: Business-driven IS Change - Illustrative Quotations of Interdependency IS Changes

Business-driven IS Change – Interdependency	Company	Informant
"We have to look across applications. You know you make one change within a policy system, well does there have to be something that goes over to the general ledgers…or how does it all feed into all the different types of reporting?"	InsuranceCo	VP of Systems Implementation
"So now we have another request to update that field but, based on a particular situation, we have code that updates that little field in a number of very different places. Well now, what was a simple change is a complex change because what if I put the code in here based on this and I get over here and the code changes?"	InsuranceCo	Senior Developer B

4.2.1.4 Implications of Business-driven IS Change Perspectives

Given the exploratory nature of this study and to facilitate reflection, it will be useful to discuss the implications of various ideas as the analysis is revealed. The intent behind the construct "Business-driven IS Change" was to be clear that while there are other IS changes taking place in applications, there is something interesting about those that are business-driven. This conceptualization ties back to the conceptual development of this research which argues

that there is a bi-directional relationship between organizational agility and the organization's information systems. This conceptualization led to the idea that it would be beneficial to better understand in what ways we can describe business-driven IS changes.

Originally, it was expected that a new way to look at IS changes would be to abstract the IS change from the applications to which they occur. As a result, we designed an interview protocol that allowed us to gather examples of business-driven IS change, from which we could abstract the general types of IS change without reference to a particular application infrastructure type or structure. The list of general types, while likely incomplete, gives a starting point for comparing general change types on different types of application structures. For example, service-oriented architecture is sold as a better structure for changing business processes, integration, and adding new functionality through creation of composite applications. As this list of general IS change types is expanded and the ability of particular application infrastructures to meet those changes is evaluated, organizations facing particular change situations will have additional evaluation points for constructing their infrastructure to match their needs. In addition, developers will be better able to identify weaknesses in their own development efforts.

There are several angles worth mentioning regarding types of business-driven IS change. First, organizations seem to have expectations of a certain profile of expected types of IS change and different organizations have different expected profiles. For example, IS departments within companies with dynamic environments may receive requests for new functionality or process changes more frequently than some other types. Second, IS departments may find that some types of business-driven IS change are easier to deal with than others. Third, some types of business-driven IS changes may be more closely linked to strategic moves by the organization.

Fourth, some types of business-driven IS changes may be easier to anticipate and prepare for than others.

Organizations may also start considering the implications of local and interdependency type changes in designing and structuring their application infrastructure. Evidence indicates that local changes are easier to accommodate in general compared to interdependency changes, which leads to the question of whether all applications should be monolithic, or place all of the functionality into a single system. This would ensure that IS changes to the application are local. The fault in this argument is that with monolithic applications, the complexity is transferred to a single application, which would result in more difficult local IS changes. Consequently, organizations should recognize that there is a balance that must be struck in structuring applications, and that evaluation of where change might occur may prove beneficial for preparing for change in the future.

Anticipated IS changes are generally easier to accommodate according to informants. While this likely has a strong design implication, informants may also have a better conceptual handle on anticipated changes because they have been previously considered, and consequently, are more equipped to handle those changes. Additionally, firms will differ in the extent to which they can anticipate changes, which implies that organizations should consider ways to anticipate IS change and prepare for it.

4.2.2 Application Characteristics Relevant to the Change

This research is interested in discovering and describing particular application characteristics that come into play during IS changes (Figure 4-3). The premise is that while there may be a wide variety of characteristics describing applications, only a subset of those are relevant to particular changes. From the interviews, a list of application characteristics that

affect, either positively or negatively, the ability of organizations to change their IS was created. In the figure, the (+) represents characteristics that are generally considered to enable IS change and the (-) represents those characteristics that generally inhibit IS change.

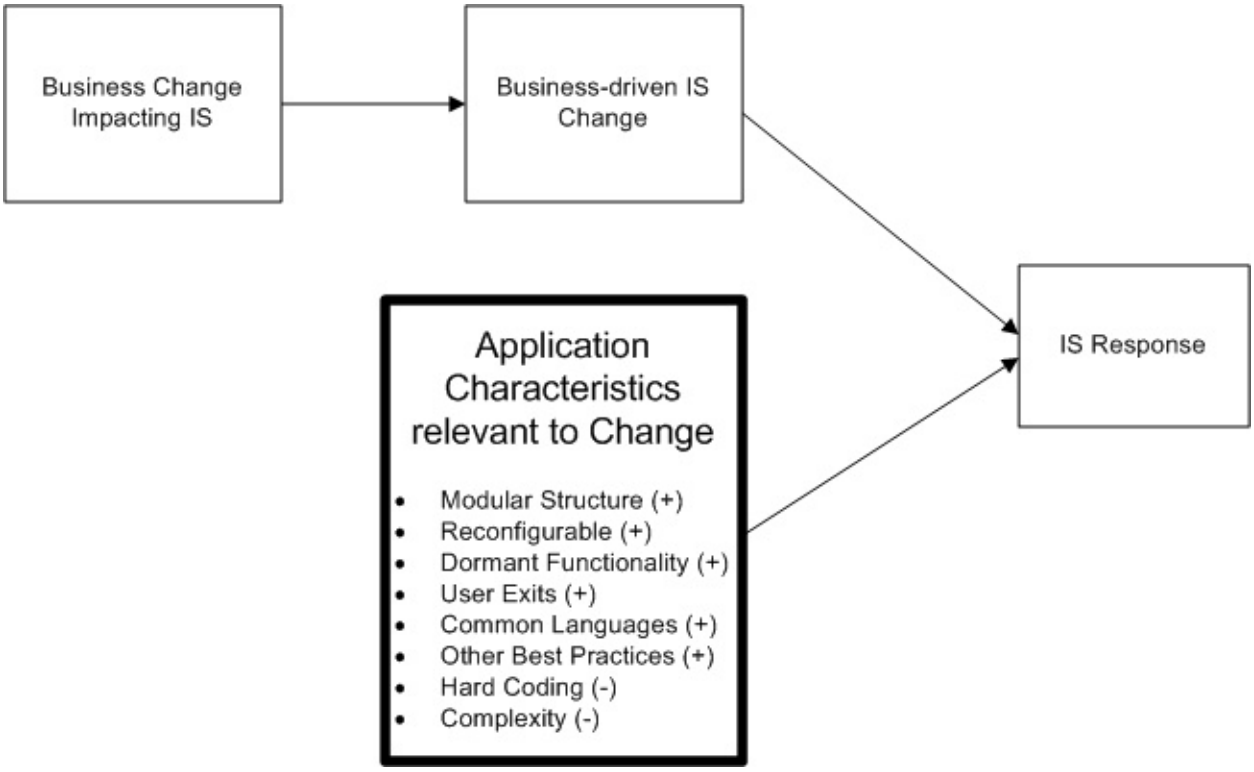

Figure 4-3: Application Characteristics Relevant to Change

Table 4-10 defines the various application characteristics relevant to change that were uncovered in this study. Table 4-11 shows the number of interviews by organization that mentioned the various application characteristics. Reconfigurable applications, including table-driven designs and configuration files, were the most common positive characteristics found, followed by modular design and user exits. Complexity – interdependency and complexity – not hidden, along with hard coding were the most common negative application characteristics relevant to change. Again, it should be noted that all of these characteristics were not planned or

desirable. Some emerged, were by-products of the design, or evolved from changes over time. Nonetheless, they are all characteristics that, in the end, have the potential to affect IS response ability.

Table 4-10: Working Definitions of Various Application Characteristics relevant to Change

Application Characteristic	Definition	Effect on IS Response Ability
Modular Design	The construction of software code into logical segments for organization and management purposes. This may be done at object level, service level, or other	+
Reconfigurable	The construction of software code which abstracts variables, making those variables easily changed. Those variables can be changed to modify the operation or properties of the application. Typically, reconfigurable structures utilize database tables to specify variable values	+
Dormant Functionality	The addition of functionality into an application which is not needed at present, but may be at a later date	+
User Exits	Places in software that allow for the opportunity to leave one application and interact with a second application, then return to the original application with perhaps some changed values	+
Common Languages	Standardizing so that applications are built with software languages that are designed to work together. Developers' skill sets are more easily transferable.	+
Hard Coding	Inclusion of variables within the application code that requires code modification to allow change (as opposed to reconfigurable)	-
Complexity - Hidden	Utilizing structure of the code to separate complexity from areas that need to change (unspecified modularization)	+
Complexity - Not Hidden	Complexity associated with the structure of the code where the complexity is mixed with areas that need to change, and thus presents more difficulty for making changes	-
Complexity - Interdependency	Complexity associated with the interconnections between applications or modules	-

Table 4-11: Interviews Indicating Various Application Characteristics Relevant to Change by Organization

Application Characteristic	RetailCo (of 5)	InsuranceCo (of 5)	SoftCo (of 5)	PowerCo (of 5)	BankCo (of 1)	Total (of 21)	Percentage (of 21)
Modular Design		1	1	3		5	24%
Reconfigurable	2	3	3	2		10	48%
Dormant Functionality						0	0%
User Exits		3	2			5	24%
Common Languages		1		1		2	10%
Hard Coding	4	3	1			8	38%
Complexity - Hidden			1	2		3	14%
Complexity - Not Hidden	1	3	2	2		8	38%
Complexity - Interdependency	4	2	3	3	1	13	62%

The mix of characteristics, both positive and negative, affects the difficulty of IS change for organizations trying to manage their application infrastructures. While this list is not exhaustive, it serves as a starting point for recognizing those characteristics of applications that ease or inhibit IS change in organizations.

4.2.2.1 Evidence of Various Application Characteristics

The first application characteristic relevant to change is modular design (Table 4-12). Modular design is a technique where programming components are separated in an attempt to simplify, organize, and compartmentalize pieces of code. It may occur at different levels, such as objects, services, expansion modules and plug-ins, or other levels. This research does not distinguish between the levels, but future studies could differentiate between different levels and their effects on change.

The goal of modular design is to build modules that have high interactions within modules and low interaction between modules. Using this structure, changes made to one module has few effects on other modules.

Table 4-12: Application Characteristics – Illustrative Quotations Related to Modular Design

Application Characteristics – Modular Design	Company	Informant
"I'm going back to the way that the whole system itself, as being a multi-layered component based system, and each piece is sort of atomic, we took the reusable approach. Like, here's the piece that puts it on the queue, here's the piece that reads the message off the queue, here's the one that decides if I delete or add or whatever. I believe that if we had built it in an older style of application development, we could not have made that change, we would have literally had to tear the code up and restart it."	PowerCo	IT Architect
"Technology is now that you can make calls to the database without disrupting the system. It can undergo a visual change, cosmetic change, you name it, as long as the data is still where it is at, internally it allows us to integrate a whole lot easier."	PowerCo	Business Analyst A

Organizations are also taking advantage of reconfigurable designs when responding to IS change needs (Table 4-13). Reconfigurable designs use either configuration files with easily modified parameters to modify application characteristics or database tables where fields can be modified to affect the application. Table-driven designs extract variables from the software code to eliminate hard-coding in particular situations. Following are some examples of reconfigurable designs described by the informants.

Table 4-13: Application Characteristics - Illustrative Quotations Related to Reconfigurable Design

Application Characteristics – Reconfigurable Design	Company	Informant
"The business also with Insight has flexibility to update the support tables. It's much more table-driven, so if I want a new value on a field I can add it to a table and it will automatically populate the database and move it through the system."	InsuranceCo	Senior IT Manager
"…we had written it originally so that it would go to individual adjusters. But the fact that we had made where it was assigned a table-driven function, we were able to add storm centers to that easily and functions into the program that allowed those to be dispersed among the multiple adjusters easier."	InsuranceCo	Senior Developer A
"…so when changes are made to a tax code or city ordinance that would affect billing to a customer, those are easily processed. We are proficient at making those changes. We are very flexible in allowing those changes. We can make a change to a database table pretty quickly."	PowerCo	Business Analyst A

Another application characteristic that was discussed in interviews was dormant functionality. Dormant functionality is designed into an application, but is "switched off" until it is needed. This allows organizations that have planned ahead for particular needs to easily enable that functionality in order to meet IS change needs. Analysis of the interviews did not indicate any examples of dormant functionality in actual IS change examples (Table 4-14), but it is expected that it is a feature that can come into play because evidence was found that dormant functionality was being built into some applications in preparation for change (Section 4.3.4).

Table 4-14: Application Characteristics - Illustrative Quotations Related to Dormant Functionality

Application Characteristics – Dormant Functionality	Company	Informant
No specific references to using a dormant functionality to respond to a stated business need. See Section 4.3.4 for an example of building in dormant functionality as a design feature.		

While dormant functionality may be a useful design feature, developers recognize that it is difficult to foresee particular functionality needs in the future. Instead, they can take advantage of user exits, or application interfaces designed to allow for external connection and communication with other software (Table 4-15). This allows developers to respond to the need for IS change by using external applications or modules to change a process, add a process, or collect external data for use within the application.

Table 4-15: Application Characteristics - Illustrative Quotations Related to User Exits

Application Characteristics – User Exits	Company	Informant
"Eventually, we'll want to get client data from other sources – from customers directly – let a customer come in and change his own address. So we anticipated that and the coding of that has all that in mind. All those hooks available for grabbing other data from other sources down the road."	InsuranceCo	VP of System Implementation
"…their system does not offer postal discounts, zip plus four postal, and things it does internally for us to get our discount. They had no product for that. And they don't offer a real print solution. What they offer is an interface to an existing print solution which is what we took."	InsuranceCo	Senior IT Manager
"With our claims tracking…they needed to able to get the zip plus four on the addresses to group claims more closely so that they didn't have the adjuster driving over here and over here. So the zip-four was a way to group addresses. And we contract out to a third party company to validate our addresses and get the zip plus four back in. And we use a web service interface and it fits right into the middle of our program. And we go out and pull the zip four, pull it in, and we save that off in our databases."	InsuranceCo	Senior Developer A

Where other techniques are not used (for whatever reason) or in addition to other techniques, common programming languages among applications provide a number of benefits for knowledge reasons among IT personnel. Technically, common languages simplify compatibility issues between applications and allows for IS change to be performed more easily. In fact, the lack of common languages can be a great inhibitor to IS change as illustrated by the first quote in Table 4-16.

Table 4-16: Application Characteristics - Illustrative Quotations Related to Common Languages

Application Characteristics – Common Languages	Company	Informant
[This system] was written in C and it was designed to be called by another program. The caller would pass in certain information and this one would send some back. Well what I was limited by there was the difference in the programming languages…My system had been written in COBOL but I needed to call a program that was written in C. And back then…you didn't have the "just make a Windows DLL out of it" option that you do now. I actually had to embed C inside of a COBOL program and I couldn't find anyone that knew how to do that."	InsuranceCo	Senior Developer B
"I will tell you one decision we made, and I told you we are a Microsoft shop. I think in 2001, we had a C [language] change and we said we are going to do C# [sharp] and that's all we're going to code in… C# is as close to Java as I could get so I am hedging my bets. Because I realized a lot of systems that we buy on the external market are Java based, and I will eventually have to support those, or do some kind of connection with them. That one decision has really helped almost as much as the n-tiered solution. Everyone has a common language now that they can communicate in, all of our developers. We found we were organized in a portfolio model, we have a team that supports marketing's applications, a team that supports accounting's side of the house. And used to, those were pretty isolated islands. Today in crisis, I can move someone from one portfolio to another. Although they may not know the business logic, they can look at the code, and say I understand the model, I understand the language that it's written in, if I have to solve a business problem, I can do that. We've seen a lot of benefits to that standardized kind of stuff that I didn't think we we're going to make. I was really trying to hedge my Java bet at the time, but it turned out to have a lot of downstream benefits that I didn't think of at the time."	PowerCo	IT Architect

Not all of the application characteristics relevant to change are positive though. Informants also discussed several dimensions of complexity as characteristics of applications that influence their ability to change. Complexity affects the ability to change by requiring more effort. In other words, if the area that must be changed to meet the IS change need is complex (Table 4-18), then it is harder to accommodate. On the other hand, if the complexity of the system is hidden (Table 4-17), perhaps within modules, or in another area of the system, then the IS change may be more easily performed. Accordingly, application structure has an effect on the difficulty of change by hiding the complexity (by separating complexity from the code that needs to be changed) or segmenting the application into easily understandable pieces.

Table 4-17: Application Characteristics - Illustrative Quotations Related to Complexity - Hidden

Application Characteristics – Complexity – Hidden	Company	Informant
"Depending on how the code was written, it could have been segmented to where complexity was separate from the things that change a lot, which could have made it much easier to change. But if you've mixed up the complex stuff with the changes you expect, it takes a lot more effort and knowledge to untangle."	PowerCo	IT Program Manager
"It's complex for the builders, it's simple for the consumers [other developers]. I have a group of people that are really technical, they are sort of the in-house help desk for our developers, and they build a lot of these, what I consider, infrastructure type components, and I say all the time, I don't care if it's hard for you, it has to be easy for the people who are going to consume them."	PowerCo	IT Architect

Table 4-18: Application Characteristics - Illustrative Quotations Related to Complexity - Not Hidden

Application Characteristics – Complexity – Not Hidden	Company	Informant
"Part of the problem, however, that makes it complex and makes change difficult is a lot components, a lot of pieces and parts. And if any one of those pieces fail because of the complexity of the system as well as the size…then our system fails."	RetailCo	System Manager A
"…our finance systems, which is probably the oldest we have…the table and the column names, they are just a bear to understand. You basically get five characters to figure out what that is and those types of things. So getting in there and learning that to make a change is very tough."	SoftCo	Business Analyst A

Also related to application structure, but in a broader sense is complexity associated with interdependency (Table 4-19). Interdependency complexity is complexity related to the relationships between different applications. Depending on the evolution and acquisition of the various applications in the organization, integration efforts often result in many point-to-point connections with data and process dependencies that may result in unforeseen consequences if an IS change is performed on even one of the applications. This was a problem found in all of the organizations included in this study. Several informants inquired jokingly if the "spaghetti chart" showing the interactions between all of the applications had been shown to the interviewer by others.

Table 4-19: Application Characteristics - Illustrative Quotations Related to Complexity - Interdependency

Application Characteristics – Complexity – Interdependency	Company	Informant
"From a technology portfolio, we have really a hodgepodge of web, client/server, UNIX, SAP, and middleware. We have a very, very challenging environment to develop in and the other thing, we have a business warehouse as well…So we have a lot of constraints or interfaces that we take into consideration whenever we make a change."	RetailCo	System Manager A
"If I've got to change a particular piece of an application that relies on a component that is used in a lot of different places and it's the component part of that piece that needs to change, that makes the change more difficult. If I've got to change…a piece of data that's being set and passed to a table at the time an application of a business process submits it and that piece of data is set in a large number of different and unrelated places, that makes the change more difficult."	InsuranceCo	Senior Developer B
"We had a lot of point-to-point integrations and it becomes very expensive when you want to change something like the balance, that you want to show on a string whether that's in a branch channel or your internet channel or your voice channel or your VRU channel call center, etc, ATM. So we have a lot of point-to-point integrations between the host and those channels. So when you wanted to make a change you had to make that change in five or six places in order to see that change come to fruition. So you couldn't just change it in one place and everybody get it."	BankCo	SOA Manager

Finally, informants pointed to hard-coding in systems as an inhibitor to IS change (Table 4-20). Hard-coding is found primarily in older applications, but that is not the only place. Even more recently developed software may have this problem. When hard-coded variables or processes are present, developers generally must make changes at the code level because simpler reconfiguration changes are not possible.

Table 4-20: Application Characteristics - Illustrative Quotations Related to Hard-coding

Application Characteristics – Hard-coding	Company	Informant
"[The legacy systems] were not built with flexibility in mind. They're all COBOL mainframe systems so if you want to add another state or another line of business, there's a whole lot of hard coding you're going to need changed. There's a whole lot of…so globally as an IT organization, those systems make that kind of business change tough."	InsuranceCo	IT Development Manager
"[MyInsure, a JAVA program] has a lot more hard-coded features as far as the wording goes…It wasn't designed really with upgrades in mind so…if we add a new line of business, they don't plug into it as easily as it should. Better coding would have allowed pieces to be plugged in for additional lines of business – to be added more easily."	InsuranceCo	Senior Developer A
"So my unit of measure, fundamental unit of measure changes. The way I pay against that unit of measure changes and where I pay. Because before it was national, everybody paid the same thing, it was buried in the code. Now you've got more exceptions: this thing gets that thing, this thing gets that thing, and this thing gets that thing…The fact that when you have a lot of hard coded type of functionality that sits behind there as well as in the fundamental underpinning that starts to be challenging."	RetailCo	Director of Business Process and Technology

4.2.2.2 Implications of Findings Related to Application Characteristics

Even as organizations choose to use applications with modular designs and reconfiguration options, they may be faced with increasing complexity over time as a result of adding new functionality, automating more aspects of the business, making more data readily available to decision makers, and complying with regulations. An interesting question is how should organizations deal with increasing complexity in their systems that may result from making changes over time. One option is for organizations to focus as much as possible on increasing those application characteristics that ease IS change, either through product selection or development efforts.

Modular design and reconfigurable services are becoming quite prominent in emerging service-oriented architectures (SOA) and business process management (BPM) efforts. Organizations are beginning to think more about modularizing around high level business services that can be rearranged and grouped together with the goal of quickly and easily making changes to business processes. As SOA and BPM become more common in organizations,

architects and developers will need to think more about the effectiveness of the application characteristics at meeting their change goals.

4.2.3 IS Responses

From the initial conceptualization of this research, there were three IS responses expected to arise from the combination of Business-driven IS Change and the Application Characteristics Relevant to the Change (Figure 4-4). These were split into Types I, IIA, and IIB. A Type I response was a response to foreseen and prepared for environmental diversity. Logically this makes sense in that if a particular change is foreseen, then the application can be built to more easily accommodate that change when it occurs[10]. In practice, this response is typically a reconfiguration change. The Type II responses are responses to unforeseen environmental diversity. In general, Type II responses involve recoding (Type IIA) or building/purchasing a new module or application (Type IIB) to meet the IS change need.

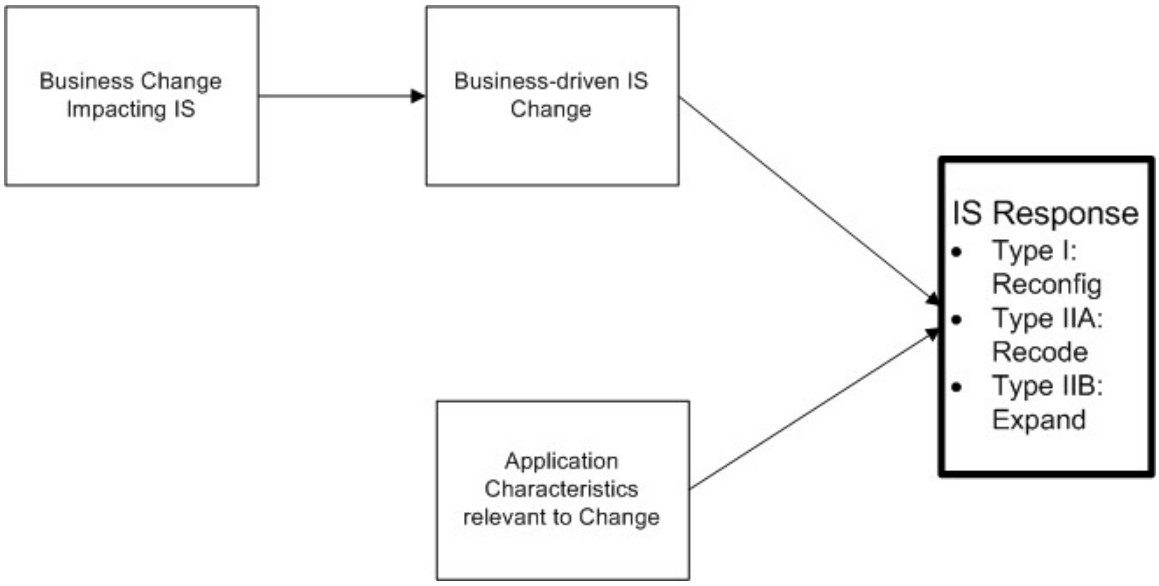

Figure 4-4: Types of IS Responses

[10] Again, there is the possibility that a particular IS change is foreseen, but no measure is taken to prepare the application to accommodate it more easily when that change is necessary.

Evidence of each of the three expected types of IS responses (Types I, IIA, and IIB) were found in 4 of 5 organizations (Table 4-21). In general, Type I (reconfiguration) responses were the most easily accommodated, both in speed and cost, demonstrating value in those cases where a particular IS change interacts with the appropriate application characteristics. From another perspective, it is apparent that responses to foreseen changes represent only a fraction of the total IS responses. In other words, recoding and expansion responses were found to be quite common, which indicates that many IS changes cannot be foreseen, and therefore, require more significant effort to accommodate. This coincides with another study which states that two-thirds of all changes to applications were unexpected (Goodhue et al. 2008).

Additionally, a new variation on the recoding response emerged from the interviews. In a few examples, informants indicated that recoding responses allowed for the opportunity to also restructure the program to simplify the application or prepare for other potential changes in the future. This is a new wrinkle and will be discussed in more detail later in this section.

Table 4-21: Interviews Indicating Various Response Types by Organization

IS Response Type	RetailCo (of 5)	InsuranceCo (of 5)	SoftCo (of 5)	Power Co (of 5)	BankCo (of 1)	Total (of 21)	Percentage (of 21)
Reconfiguration (Type I)	1	3	5	3		12	57%
Recoding (Type IIA)	3	4	5	4		16	76%
Recode and Restructure	1	2				3	14%
Expand (Type IIB)		3	3	3	1	10	48%

4.2.3.1 Evidence of Various IS Responses

Type I responses, or reconfiguration responses, are a result of the union between a particular IS change and the application that has been "prepared" for that change. In many cases, Type I changes are a result of good planning or good programming practices related to anticipation of changes (more discussion of this issue in following sections). In other cases,

reconfiguration may involve enabling dormant functionality, or in emerging business process management (BPM) systems, rearranging software modules or services. In general, Type I responses are the most desirable way to accommodate the need for IS change. Table 4-22 shows some informant experiences with Type I changes.

Table 4-22: IS Response - Illustrative Quotations Related to Type I - Reconfiguration

IS Response – Type I - Reconfigure	Company	Informant
"When I say it's data-driven, for example in CPS, if you wanted to add an additional line of business and you didn't have any special edits, you could put some entries into what we call the PVT [a table] and they'll start appearing. Those payments will start appearing to be due and you can handle them in that process because it was very data-driven and it was planned for."	InsuranceCo	IT Development Manager
"We put in billing plans that had a particular service charge attached to every invoice. And we made a business decision, I say we, the underwriting made a business decision and said look we need to encourage people to go to electronic funds transfer. So let's give them a lower service charge. So I worked with the business community to debate and argue about what that was - did the financial analysis and so forth. They submitted this and this is an extremely simple change. It's a configuration change. And then figure out how to convert the customers that are already on an old bill plan. So the technical part of this is simple. The hard part was getting the business community to agree on the dollar amount."	InsuranceCo	VP of Systems Implementation
"You go in and you select what rule you want to change. You edit it, modify it, alter the conditions, add this rule if this product exists and this product exists and/or if they're a new client. You can do all kinds of rules. We do the same thing with like territories of who can sell for what in the application."	SoftCo	Business Analyst B

When reconfiguration is not an option, IT departments have a few other options to meet IS change needs. First, they may choose to recode the application (Table 4-23). Recoding can be relatively small or quite extensive and is affected by both the extent of the particular IS change and the application being changed.

Table 4-23: IS Response - Illustrative Quotations Related to Type IIA - Recoding

IS Response – Type IIA – Recode	Company	Informant
"Well our systems need to be changed to accommodate that new plan that wasn't there before. So you've got your code change for that."	RetailCo	Director of Business Process and Technology
"So we had to write additional functionality into our system for the catastrophe situation so that it was easier to disburse the claims among multiple adjusters and handle bulk claims in one area."	InsuranceCo	Senior Developer A

An unexpected variation of the Type IIA response is the recode and restructure response (Table 4-24). At times, developers take the initiative to improve the structure or organization of an application when an IS change is required. Essentially this is an acknowledgement of the difficulty of change to complex applications which results in not only recoding to meet the IS change need, but also entering an anticipation mindset, where future changes are being anticipated and incorporated into the application.

Table 4-24: IS Response - Illustrative Quotations Related to Recode and Restructure

IS Response – Recode and Restructure	Company	Informant
"Well now what was a simple change is a complex change because what if I put the code in here based on this and then I get over here and it changes? And what I had to do in that case was I had to write a different request that said, you know, take all of these various places where that code is evaluated and put them in one spot. And then we'll evaluate this new change at the same time so that what goes in that field then is in one place and so it gets modularized."	InsuranceCo	Senior Developer B
"That is hard coding. That says if it's equal to quote 'this', do so and so. Well anytime that we have an opportunity now you know to table those, to put that stuff in a table that we can read and then modify so that there's no attempt or need to reissue the application or release the application because we had to make a hard coded value change, we learned from that years ago."	RetailCo	System Manager A

Type IIB responses, or expansion responses, involve the addition and integration of new applications or development of significant new modules or services to meet IS change needs (Table 4-25). There are a number of situations where expansion responses are perhaps more beneficial than recoding existing applications. First, when the IS change would require

significant recoding of existing applications, organizations may find it simpler, less expensive, and faster to purchase or separately develop a new piece of software, leaving intact a working application. Second, when organizations have limited resources for development, acquiring a new application and integrating it with existing applications is more feasible. Third, for organizations employing emerging software designs, such as service-oriented architectures and BPM, services or modules may be more economical and more versatile in the long run than recoding.

The IT Program Manager at PowerCo describes their decision making regarding expansion:

> "If we don't have an existing solution, we would seek out 3rd party solutions primarily for seeking best practices and leveraging economies of scale, and for balancing the support requirements long-term. 3rd party solutions can become very expensive because of the maintenance, and having to upgrade, and you're not as flexible. So there are tradeoffs."

Table 4-25: IS Response - Illustrative Quotations Related to Type IIB - Expand

IS Response – Type IIB - Expand	Company	Informant
"We were previously on a single car policy and we converted to multi-car. And the single car policies in legacy all had different due dates. And for regulatory reasons, we can't just cancel them all as of a certain date and rewrite them in the system. So we had to write a program where as they came up for renewal, they converted in the system."	InsuranceCo	VP of Systems Implementation
"…so what we did is we built a system, a custom application, that allows the users to create pricing in one system, a master record, and then pushes those changes to the other systems automatically."	SoftCo	CIO

A final IS response is to replace the application entirely. While this option is newly added to the IS responses, it should be recognized as a possibility for dealing with business-driven IS change needs. In some circumstances, organizations may find that IS change needs are so drastic and existing applications are so outdated, that replacement of the existing application outright makes sense as a way to meet business-driven IS change needs. Generally it is expected that this option will be chosen when the IS change need cannot be accommodated by the other IS

change alternatives and the urgency of the change is lower. In the interviews, there was no clear evidence that this was done in response to specific IS change needs. In a couple of instances, however, informants acknowledged new applications, such as an ERP to replace the old financial application, were being implemented. It is assumed that this is driven by substantial IS change needs, and rather than recoding or expanding an old application, a new one was chosen to replace it.

4.2.3.2 Mismatch between IS Change Request Difficulty and IS Response

Interestingly, in a number of examples, informants discussed that there is at times a mismatch between the difficulty of the IS change request as seen by the business and the difficulty of the IS response as experienced by the IS department. This further emphasizes the relationship between the particular IS change and the application characteristics relevant to the change as contributors to IS responses. In other words, if the interaction between the particular change and the application characteristics relevant to the change were not a substantive issue, then one would expect that there would consistently be a parallel between the apparent IS change request difficulty and the actual IS response difficulty.

To illustrate this point, examples for both simple and difficult IS responses are excerpted from the interviews. In some cases, the business people within the organization request IS changes that appear to be simple to accommodate. However, responding to that IS change is just the opposite. Table 4-26 gives two examples.

Table 4-26: Mismatches - Simple IS Change Requests and Difficult IS Responses

Mismatch between IS Change and IS Response	Company	Informant
"…this is just a great nightmare for me . . . we found that there are some policies that if they cancel because they don't make that last payment on the monthly payment plan and then they come back in just prior to the renewal date and say I'm sorry, forgive me I want to pay and reinstate the policy. Well most companies would just reinstate you with the lapse and take your money and you basically go back to your old expiration date on that renewal. With our company, we don't want to do that. We want to issue them a new policy for a lot of business reasons that make us different from everybody else. The system won't let you do that. So we had to push through a change request to allow us to, even if that renewal arose, cancel it off and then put it back on there and let us determine what the dates are. I paid [a lot of money] to get that change in. The business community doesn't understand why that is so big. They said what a stupid system that won't do that. But technically what you're doing is canceling off a renewal row and then wanting to put it back on there but a little different and we don't want to go in and re-enter any information. It creates commission problems, it creates statistical reporting problems. Business people don't care. So technically that is a huge project for us and the business people finally wore us down and we agreed to do it."	InsuranceCo	VP of Systems Implementation
"I could give you an example of one that wouldn't, on the surface, look like it would be very complex but actually turns out, it turned out that it was. There's a particular code in one of our programs, in our cash processing system, that the mainframe makes a lot of different decisions on the basis of what goes in that field. And at the beginning it was a very simple, if this . . . if A and B, put C in the field. Well over time there were a number of other situations that required us to put different codes in that field and what happened was each time somebody else touched the program, the logic for updating that field went in a slightly different place. So now we have another request to update that field but . . . based on a particular situation but we have code that updates that little field in a number of very different places. Well now what was a simple change is a complex change because what if I put the code in here based on this and then I get over here and it changes? And what I had to do in that case was I had to write a different request that said, you know, take all of these various places where that code is evaluated and put them in one spot. And then we'll evaluate this new change at the same time so that what goes in that field then is in one place and so it gets modularized."	InsuranceCo	Senior Developer B

In other cases, the business people request IS changes that seem quite difficult to accommodate, and perhaps expect long lead times and to pay extensively for the change. However, if that application has the appropriate characteristics for accommodating that change, then the IS response is quite simple (Table 4-27).

Table 4-27: Mismatches - Difficult IS Change Requests and Simple IS Responses

Mismatch between IS Change and IS Response	Company	Informant
"We pay so many windshield claims that we wanted to give a glass company access to come in and directly look at our policy records in a very limited way. And we could control what every user sees on the screen. So the claims department requested that. They felt like it would be a year long project, it was simply a configuration. We just went in there . . . and security issue, so we got security people, configuration people, and we had that in and tested within a couple of weeks."	InsuranceCo	VP of Systems Implementation

4.2.3.3 Implications of IS Change Responses

Evidence from the interviews confirmed that type I reconfiguration responses are the most desirable for organizations dealing with the need for IS change. While that is no surprise, looking through the lens which considers both the particular IS change and the application characteristics relevant to the change emphasizes that organizations do have some control over their ability to accommodate business-driven IS change needs easily. While the effect of existing applications on meeting particular IS change needs has already been determined at the point where they were designed and implemented, future application selection or development can be made with more consideration of the features available to accommodate change and which potential IS change needs might arise.

Additionally, the uncovering of recode and restructure response led to a key piece of evidence that organizations are not tied to their fate, so-to-speak, having chosen and implemented their application infrastructure. While recoding meets the current need for IS change, the restructuring allows for organizations to enter into an anticipation phase where they can attempt to predict and, at least to some extent, prepare for future IS changes.

In fact, the recode and restructure IS response is an acknowledgement that organizations are aware of the environment in which they are performing their jobs. They recognize the IS changes are the norm now for organizations and they can potentially make future changes more efficient, which benefits both the group responsible for the changes as well as the organization as

a whole. This mentality represents a collection of interesting new findings from the interview data related to the anticipation of changes to the IS.

4.3 ANTICIPATION OF CHANGE TO THE IS

While it was acknowledged in the initial conceptual model that previously anticipated changes to the IS impacted the application characteristics that, in turn, influence the ability to change when change is necessary, it became quite apparent from the interviews that the extent to which companies anticipate and prepare for change varies considerably across firms and situations. More specifically, we found evidence that there are strategies to change anticipation, targeted and general, and anticipation tactics in place to prepare for forthcoming changes. Additionally, there is evidence that perceptions of anticipation ability and anticipation benefits affect the strategies and tactics utilized.

An example from Senior Developer 2 at InsuranceCo illustrates the core ideas of anticipation strategy and anticipation tactics. Initially, the informant acknowledges the anticipation of a particular business change – a targeted anticipation strategy.

> "When you're building a program you know that six months or a year from now or put in some sort of time frame, you're going to have to change the rating part of it."

The anticipation of this particular business change leads to an anticipation tactic – making the application reconfigurable by separating variables that are expected to change from the code and placing them in a table-driven structure.

> "…there may not be really any code changes as far as how it calculates a rate but your data tables may change – your base rates that it uses. And so you try to write the program so that you can easily update your rating tables or your base rate tables without having to necessarily impact the calculation routine."

By preparing for potential change to the calculation routine and modularizing it to separate it from impacting the interface to the program, the developer is also using a second

anticipation tactic – modular structure. As a result, modular structures can aid in hiding or reducing complexity encountered when IS change is necessary in the future.

> "And you try to put the calculation routine off in its own little world so to speak, you try to make it pluggable so that I could change the calculation routine without changing the interface to the program."

While not stated directly in this example, the implication from the developer is that he perceives benefits from implementing change contingencies into the application. Additionally, the developer is acknowledging some confidence in the ability to anticipate a business change in this case.

In a continuing example from the IS change and IS response discussion (Section 4.2), the CIO of SoftCo discussed how the integration application they developed also included anticipation of future changes, whether it was a pretty certain change like converting their financial system to SAP or a less certain change like integrating unknown systems to the mix.

First, the CIO acknowledges some ability to anticipate both technical changes and business changes. Furthermore, the CIO mentions both targeted strategies, such as planning for the forthcoming SAP conversion, and general strategies, including using web services technology for integration possibilities in the future.

> "…the translation environment that I mentioned is malleable if you will. That is if we added another system, we could sort of add that system in as another value within the translation environment. Or if we retired one we could just remove it. And on the integration layer, meaning the technology between the systems, we use web services for integration kind of activity. So, for example I mentioned we're replacing our financial system with SAP which is actually…a system of record for these SKU codes."

Based on the anticipation strategies, the price integration application incorporated web service interfaces to make reconfiguration for the SAP conversion and future applications simple.

> "So what we're going to do in the current system where this SKU code resides, the master file is going to be replaced and we're going to put SAP in and we'll let the integration to the other five systems remain unchanged. Well there might be some minor modifications but the point being that with the web services integration to the financial system, you can swap out the financial system and put another one in and these other hosts theoretically don't know it. Obviously I'm sure there will be something that comes up that requires a change but it won't mean tearing down and rewriting the whole system. It just means changing the interface just to SAP. So there's obviously effort cost for that but it beats the heck out of rewriting the whole thing for a six way integration project. And all of the decisions were consciously made and were part of frankly the justification for the investment that we needed to do this anyway and knowing that SAP was coming we might as well go ahead and do it now, get that piece done and test it so we don't have to worry about it and that's one less thing to worry about when the SAP project comes along."

Based on analysis of the examples from the interviews similar to those just discussed, a new model with four major constructs has been developed (Figure 4-5) to capture the ways in which anticipation can affect the application characteristics relevant to future change requests. Two constructs emerged from the interviews related to perceptions of the individuals. First, individuals have perceptions of their ability to anticipate changes. Second, there are perceived benefits to implementing anticipation tactics into applications. These perceptions affect the change anticipation strategy, which may be targeted, general, or non-existent. The strategy taken affects the change anticipation tactics, which are structures or features incorporated into the design of the application to accommodate change needs at a later time.

It should be noted that evidence for these two new constructs (perceived ability and perceived benefits) come from individual perceptions. It is unclear at this point whether such perceptions are predominantly individual or organizational.

In the following sections, each of these constructs will be explored in detail, beginning with the central construct, Anticipation Strategy.

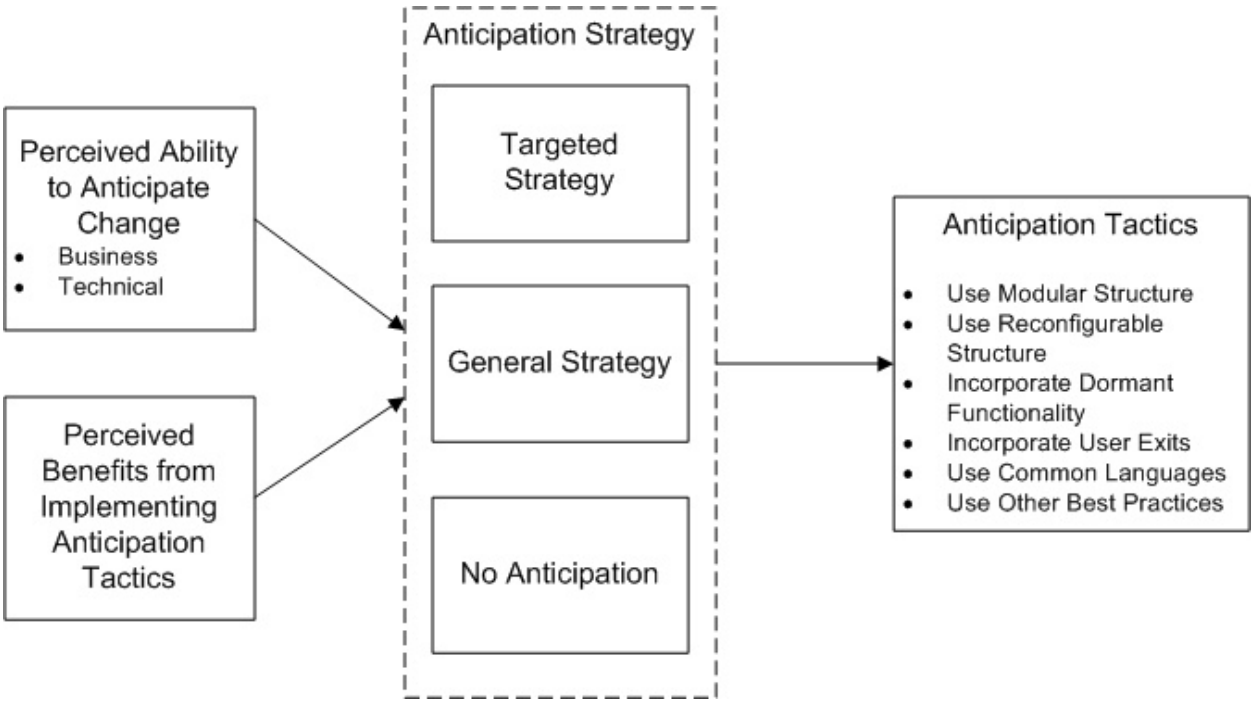

Figure 4-5: IS Change Anticipation Model

4.3.1 Change Anticipation Strategy

From the interviews, we discovered evidence that not only were changes expected for applications at a later time, but often attempts were made to anticipate those changes. In other words, changes to the IS were prepared for in a proactive sense. While each of the organizations recognized the likelihood of change at least to some degree, they varied in their approaches.

In some cases, informants described targeted anticipation. Targeted anticipation is directed at specific changes to the application and may be either from a technical perspective or a business perspective. In other words, individuals may predict specific technical changes to an application, such as shifting from a mainframe to a client/server approach, and build in the appropriate features to accommodate that change. Others may predict specific business changes, such as expanding the company nationwide, which in turn, forces technical preparation for that business change.

In other cases, informants discussed general strategies where they tried to prepare much of the application for change without using specific anticipated changes as a basis. The primary general strategy is one of broad flexibility. Broad flexibility attempts to develop applications in a way that makes the application as flexible as possible, without predicting which specific business or technical changes may occur. When developers do not have specific information about potential changes, they may still choose to abstract variables from code, use modular structures, or create user exits.

On one end of the general strategy continuum is broad flexibility as described above. On the other end are basic best practices. By using basic best practices, such as common languages, object-oriented programming, and open standards, applications may in some ways be more prepared for some future IS changes.

Finally, informants stated that, at times, there is "no anticipation" strategy. While "no anticipation" strategy may in reality be similar to the basic best practices approach, there are cases where informants stated specifically that they did not anticipate potential IS changes. Each of these strategies is depicted in Figure 4-6.

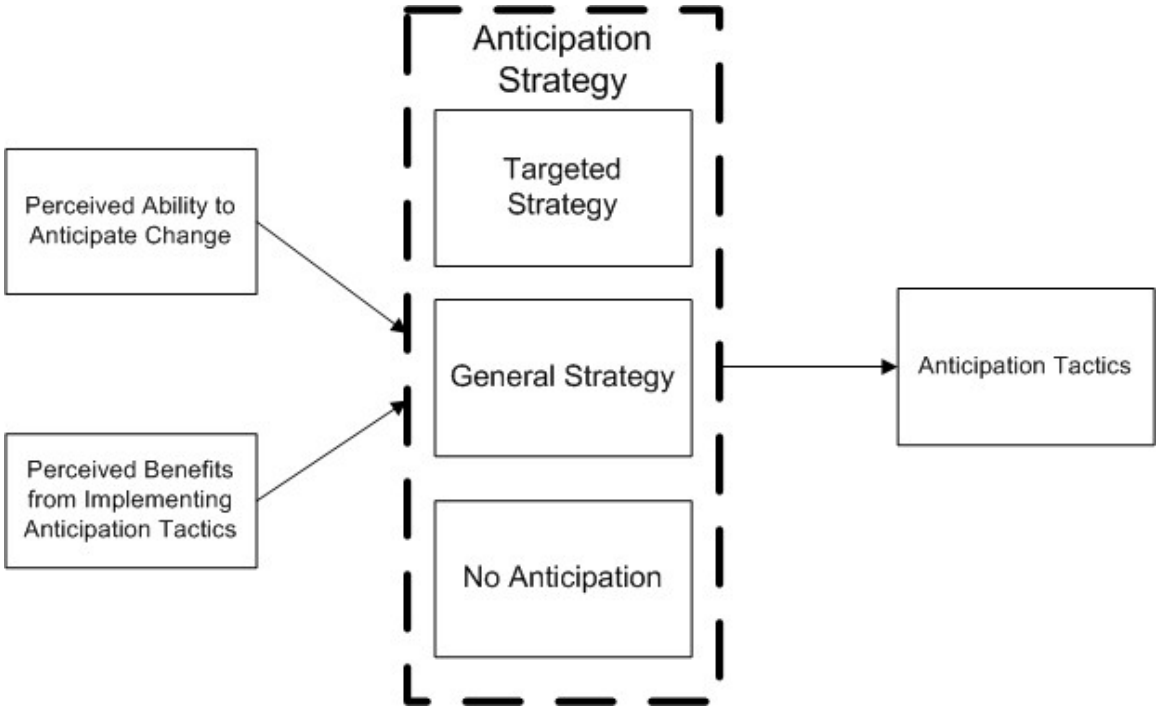

Figure 4-6: Change Anticipation Strategies

All of the organizations had some indications of an anticipation strategy (Table 4-28). Interestingly, three of those same organizations also had individuals that indicated they had "no anticipation" strategy. In fact, in two companies, one informant (at each company) indicated both proactive anticipation strategies (targeted and general) and "no anticipation" strategy. This may indicate that anticipation strategies vary not by organization, nor by individual, but by project or some finer level of detail. Still, further investigation should examine the source of this variation.

Overall, more individuals reported targeted strategies than general strategies or "no anticipation" strategy. This may reflect a preference to predicting particular changes to applications. Three companies indicated a general broad flexibility approach to anticipating change. While no evidence of the basic best practices strategy was found, it is expected that this is a real possibility in practice. For those who do not actively attempt to predict particular

changes through targeted strategies, nor attempt to just be as "flexible" as possible, a default position of adhering to basic best practices may inadvertently prepare the application for some potential IS changes in the future.

Table 4-28: Interviews Indicating Various Anticipation Strategies by Organization

Anticipation Strategy	RetailCo (of 5)	InsuranceCo (of 5)	SoftCo (of 5)	PowerCo (of 5)	BankCo (of 1)	Total (of 21)	Percentage (of 21)
Targeted - Business	1	4	2	1	1	9	43%
Targeted - Technical	2	2	1	2	1	8	38%
General - Broad Flexibility	1	3	1			5	24%
General - Basic Best Practices						0	0%
None	2	1	1			4	19%

4.3.1.1 Targeted Anticipation Strategy

A targeted anticipation strategy consists of preparing for specific changes to the application in the future. For example, a developer may have knowledge that the company is undergoing a particular expansion in the future, and consequently incorporates tactics to be prepared for that particular change. In another case, a developer may anticipate that the application will someday be accessible on the company's intranet and implement structures to accommodate that change when it occurs. Table 4-29 includes quotations related to targeted strategies.

Table 4-29: Anticipation Strategy - Illustrative Quotations of Targeted Anticipation Strategy

Change Anticipation Strategy - Targeted	Company	Informant
"So in [claims tracking] areas, a lot of the business needs that we change frequently for small stuff – 'can you change the wording on the screen?' 'We want this payment to be called this instead of that.' And with our systems, we have tried to write them to make those kinds of changes as easy as possible so that a lot of the wording on the screen is table-driven, so that it's a table change rather than a programming change" "We try to, when we write our systems, think ahead on those things to anticipate where our most frequent business changes will be. 'Who's going to get what email?' We're not going to program that. We're going to make it more table-driven." "We have to make a conscious effort to think, we're going to 50 states some day because we need to grow this company. So I have to remember and I remind junior programmers because they tend to get tunnel vision – 'here's where we are and here's what we need to program today.' And it's no, we are growing this company…How are we going to handle additional states? How are we going to handle when we get so many agents? How are we going to handle it if we need to add new lines of business?"	InsuranceCo	Senior Developer A
"With the regulatory changes, if you've got something that's doing a rating …If when you're building a program, you know that six months or a year from now…you're going to have to change the rating part of it."	InsuranceCo	Senior Developer B
"Architecting for change, even in those situations where you're loosely coupled, anticipating that today you have a web service that puts something on an IBM member queue and puts it up on the mainframe, what happens if they move it off the mainframe. Your thing that puts it on the queue still works, but where's the queue going, and your backend and stuff like that."	PowerCo	IT Architect

Targeted anticipation strategies also have an additional dimension. Individuals may have either a business or technical view (or both) when they are anticipating change. In the case where an individual is anticipating business changes, they are first concerned with the business level of change and the consequent technical changes. In other cases, an individual may just be anticipating technical changes. Table 4-30 and Table 4-31 illustrate this dimension.

Table 4-30: Illustrative Quotations of Targeted Anticipation Strategy from a Business Perspective

Targeted Anticipation Strategy – Business	Company	Informant
"We have a basic idea where our company wants to go. When I wrote the cash processing, I knew that initially it would be used for the field. I know that eventually they want to have pieces on the internet. I know that eventually we're going to be taking online claims. Even though the initial use of it was just in our service centers, I knew when I wrote it that there were pieces that were going to be added in and I tried to leave openings for that information to have a way in and those pieces fit into the program."	InsuranceCo	Senior Developer A

Table 4-31: Illustrative Quotations of Targeted Anticipation Strategy from a Technical Perspective

Targeted Anticipation Strategy – Technical	Company	Informant
"A lot of our new stuff is stuff we anticipated. We've got a new program coming in which we knew we were going to have to integrate with these. With [this program] though, we know it was coming but we didn't know exactly what it was going to be. So while we tried to keep things open for how it would integrate, we didn't have all of pieces to know exactly how it was."	InsuranceCo	Senior Developer A
"Generally on a technical level you can anticipate there - like we run this critical process weekly and we all know in the back of our minds that eventually they [the business] are going to want to see that data refreshed daily. And our platform as of two months ago couldn't handle that. And that's what we're working on now is . . . we're putting it onto a new platform because we need to be able to react faster and that's why we're upgrading it. To do that is requiring the migration of all these processes that used to exist on an old version of this server onto a new one so that we can run it daily and it will finish the process fast enough that the BW administrator team can load the data."	RetailCo	ETL Analyst

4.3.1.2 General Anticipation Strategy

A general anticipation strategy consists of preparing for potential changes, but not particular anticipated changes. Developers may not know what business variables may change, but still build the application in a manner that those variables are easily changed. An important feature of the general anticipation strategy is that this technique may prepare an application for changes even when the developer has no insight into the IS change that may occur.

The interviews also indicated that general strategies were likely in cases where individuals knew IS change was likely, but felt they had less ability to anticipate particular

changes. The following table illustrates the mindset behind the general anticipation strategy (Table 4-32).

Table 4-32: Anticipation Strategy - Illustrative Quotations of General Anticipation Strategy

Change Anticipation Strategy - General	Company	Informant
"We go with all or nothing, everything needs to be flexible… we're going through a period of rapid growth, extremely high investment in technology. . . we're in a position where, especially my team, needs to be focused on writing new stuff and not maintaining. And so I stay extremely focused on making it flexible so that maintenance is not an issue."	InsuranceCo	IT Development Manager
"The problem is the gap between how IS as a field of study behaves and how the business is run. And we're taught about rigidity in structure, repeatable infrastructure, and everything adhering to a set of rules. And then you can plan around those rules and say these are the factors that are going to change. And that's all well and good, but the business doesn't have those rules and whatever you need to execute at any given time should have been executed five seconds ago - so let's go, let's get it done. And that's the problem – it's trying to anticipate where they're going to make these changes. So all you can do is make your data types or however you're capturing this information, and try to make it as flexible as possible and just know that this is going to change. That's all you really know, and that hopefully they won't change the structure, maybe they'll just change some codes. And when they told you that this code would only have one value, it's going to have two values."	RetailCo	ETL Analyst

4.3.1.3 "No Anticipation" Strategy

"No anticipation" strategy is simply a non-predictive mindset (Table 4-33). In some cases, this strategy is forced where there is insufficient time or resources to plan for future changes because of deadlines for completion. In other words, the company needs the application as soon as possible. "No anticipation" strategy is not necessarily an acknowledgement of a lack of benefit or ability to anticipate change though. While this strategy may reflect a lack of foresight in some cases, it may also indicate a default mindset in cases where urgency for completion is high.

Table 4-33: Anticipation Strategy - Illustrative Quotations of "No Anticipation" Strategy

Change Anticipation Strategy – No Strategy	Company	Informant
"I don't know that any care was taken to forward-thinking changes. I think what we spent most of the time on was just blueprinting." "We're always kind of pushed for hours, but there are not a lot of initiatives, at least on the projects that I've worked on, where you build in functionality that's shut off. We understand it's on the roadmap that we might need to go back in and there may be some slight design changes…but there's not a whole lot of 'let's make it completely decoupled just in case we ever change something. Let's put a database in between it so that we can on the fly change some fields or add fields and stuff. We don't really do too much of that in any project."	SoftCo	Business Analyst A
"The whole thesis behind everything is that we're pursuing modularity or trying to get everything down in these little blocks so we can reassemble and change quickly on the fly when these ad hoc requests do come or when they do change a promotional effort. But that's the pie in the sky, that's the ideal and we almost never get there because someone will just make up a criteria that they need implemented in the next half hour and I'll make it work…We all know that from a software developer standpoint, from a nerd standpoint, we know that there's these IT things that we'd like to pursue that would make things really reusable and really sensible. But the business doesn't react that way and I think their pace is a lot faster."	RetailCo	ETL Analyst

4.3.1.4 Implications of Change Anticipation Strategy

The simple assumption is that anticipating IS change to applications is beneficial for organizations. This assumption makes sense based on the discussion earlier in this chapter of the difficulty that unanticipated changes present to organizations. Furthermore, informants indicated that there were benefits to anticipation of IS change (discussed in Section 4.3.3). However, it should be acknowledged that there may be tradeoffs to the three strategies, and there are situations where one is preferable to the others. For example, for some developers, targeting particular IS changes may be difficult due to their "distance" from the business, and time spent predicting particular changes may be wasteful and ineffective. Alternatively, the general broad flexibility strategy may be taken to the extreme, where development has difficulty moving beyond analysis or coding. Finally, for highly static systems, "no anticipation" strategy may be preferable to proactive anticipation strategies. Further research is needed to assess the circumstances where one strategy is preferable over others.

Additionally, the separation of targeted strategies and general strategies is not meant to imply that they should be exclusive of the other. In fact, in several of the organizations, individuals seemed to have a blended mindset. In some examples, they cited specific changes they were preparing for, while in other examples, they indicated that they were preparing broadly for change. While further research is needed, the blended strategy is expected to produce the best results for organizations. In other words, target those changes that can be foreseen, and prepare for any eventuality when it is reasonable to do so.

Finally, it is not clear where change anticipation strategies originate. It appears that there is inconsistency about strategies among individuals in various organizations, and at times, even within individuals. InsuranceCo seems to be the only organization that had some consistency between the IT development manager and the two senior developers. These three individuals all indicated that proactive strategies were important to their jobs, though evidence was that it is a "culture", rather than a directive from the leadership. The other organizations did not show consistency among the individuals, seemingly indicating that their strategies were individual rather than group-based. Perhaps, it can simply be explained by the individual's perceived ability to anticipate change.

4.3.2 Perceived Ability to Anticipate Change

Individuals involved in the development of applications have varying perceptions about their ability to anticipate change (Figure 4-7). This characteristic is expected to influence which strategy, or strategies, are taken to prepare for change. For example, individuals that do not believe they can effectively anticipate particular changes to applications are less likely to choose targeted strategies during development, and are more likely to opt for general strategies, or no strategy. On the other hand, individuals that do believe they can anticipate particular changes to

applications are more likely to choose targeted strategies aimed at addressing those particular changes.

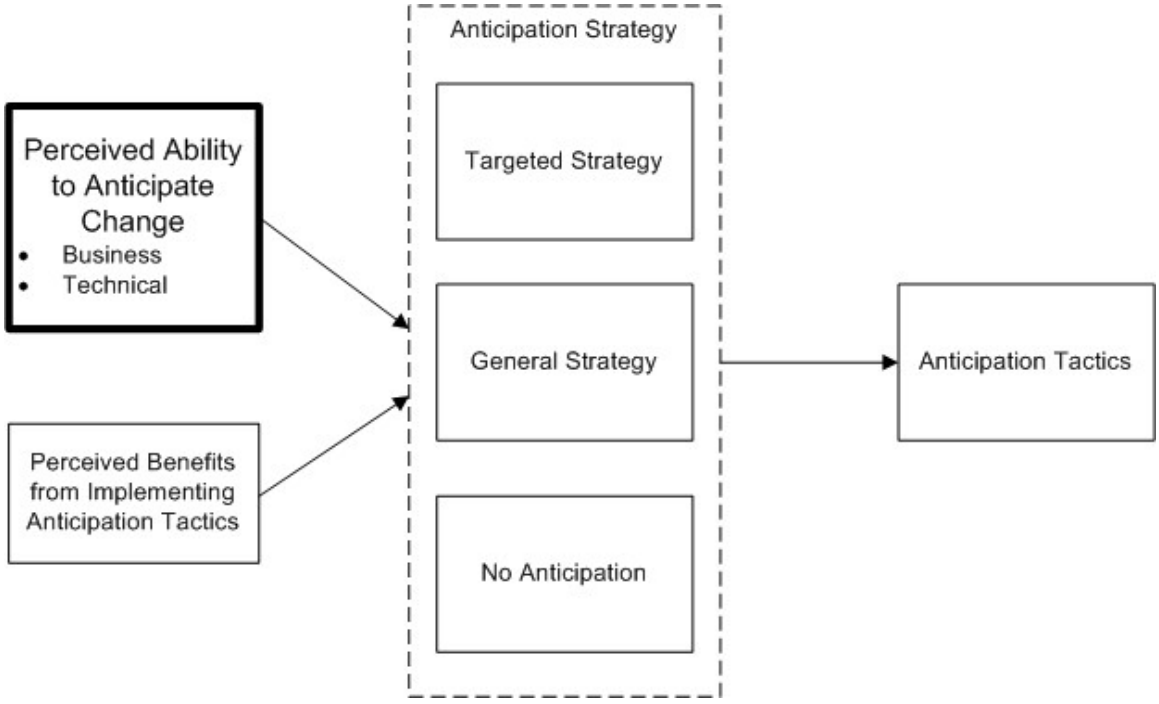

Figure 4-7: Perceived Ability to Anticipate Change

In some cases, informants indicated a general ability, either high or low, to anticipate change. In other cases, informants gave more details, which allowed for differentiation between their perceptions of whether they could anticipate change from a business or technical perspective. Overall, four individuals (one in each of four organizations) shared information regarding their ability or inability to anticipate change, and their perceptions are summarized in Table 4-34. Illustrative quotations are shown in Table 4-35.

Table 4-34: Interviews Indicating Perceived Ability to Anticipate Change by Organization

Perceived Ability to Anticipate Change	RetailCo (of 5)	InsuranceCo (of 5)	SoftCo (of 5)	PowerCo (of 5)	BankCo (of 1)	Total (of 21)	Percentage (of 21)
General Description		1	1	1	1	4	19%
Business Perspective	1			1		2	10%
Technical Perspective	1					1	5%

Table 4-35: Perceived Ability to Anticipate Change - Illustrative Quotations

Perceived Ability to Anticipate Change	Company	Informant
"There are two skills that we don't have enough of, which is hurting us. One is being able to define the business requirements clearly, and also anticipate the business requirements of the future. So it's more than just taking notes from the client at the moment. It's understanding their requirements over time…We will respond to a change and before it's over with, we've got another change coming. So anticipating that is tough."	PowerCo	IT Program Manager
"With our systems, we have tried to write them to make those kinds [screen wording] changes as easy as possible, so that a lot of wording on the screen is table-driven, so that it's a table change rather than a programming change. We try to, when we write our systems, think ahead on those things to anticipate where our most frequent changes will be."	InsuranceCo	Senior Developer A
"So you can anticipate those sorts [technical changes] of things. What you can't anticipate is the business rationale behind some of the decisions because we're just not that close to them." "And then you can plan around those rules and say these are the factors that are going to change. And that's all well and good, but the business doesn't have those rules and whatever you need to execute at any given time should have been executed five seconds ago - so let's go, let's get it done. And that's the problem – it's trying to anticipate where they're going to make these changes. So all you can do is make your data types or however you're capturing this information, and try to make it as flexible as possible and just know that this is going to change. That's all you really know, and that hopefully they won't change the structure, maybe they'll just change some codes. And when they told you that this code would only have one value, it's going to have two values."	RetailCo	ETL Analyst

Initial examination of the quotations in Table 4-35 indicates high or low perceptions about their ability to anticipate changes. A second look shows that the IT Program Manager at PowerCo and the ETL Analyst at RetailCo both indicate a lack of skill in anticipating business changes. However, Senior Developer A at InsuranceCo and the ETL Analyst both believe they have some ability to anticipate changes from a technical perspective.

Individuals' perceptions may come from any number of sources, which are currently open to speculation. Some individuals may feel that they are too separated from the business or that the business is too uncertain to anticipate changes that will affect the IS, and hence, a low perception of their ability to anticipate change. They might have little understanding of the future goals of the business, be uncertain about how it may affect the application, or lack previous experience in the business. They might have even been ineffective in the past at predicting change to the IS and consequently, have quit making the effort.

Alternatively, some individuals may feel they have the appropriate knowledge and understanding of either potential business or technical changes, and as a result, have a high perception of their ability to anticipate change. This most likely arises from experience with the business, the applications, the previous changes to the systems, or all of the above.

4.3.2.1 Implications of Perceived Ability to Anticipate Change

Again, it is expected that the perceived ability to anticipate change will affect the anticipation strategy chosen. While further research will investigate this, initial evidence shows that low perceptions of ability to anticipate change leads to general strategies or "no anticipation" strategy. High perceptions are expected to lead to targeted anticipation strategies, perhaps still in conjunction with general strategies.

This construct is an important construct for organizations. Organizations could have considerable influence over perceived ability to anticipate change by involving the IS department more in discussions about the direction of the organization. For example, if IS personnel knew about a shift in organization strategy to start looking for acquisitions, or perhaps a shift in product offerings, then IS personnel would be more certain of their efforts, as well as have the information to act upon during development.

4.3.3 Perceived Benefits of Anticipation Tactics

Along with the perceived ability to anticipate change, individuals also differ in their belief about the benefits of anticipation tactics. It is expected that individuals (or organizations) with a strong belief in the benefits of anticipation tactics will be more likely to use proactive anticipation strategies (targeted or general). In other words, if no benefit is expected from anticipation tactics, then they are less likely to be implemented.

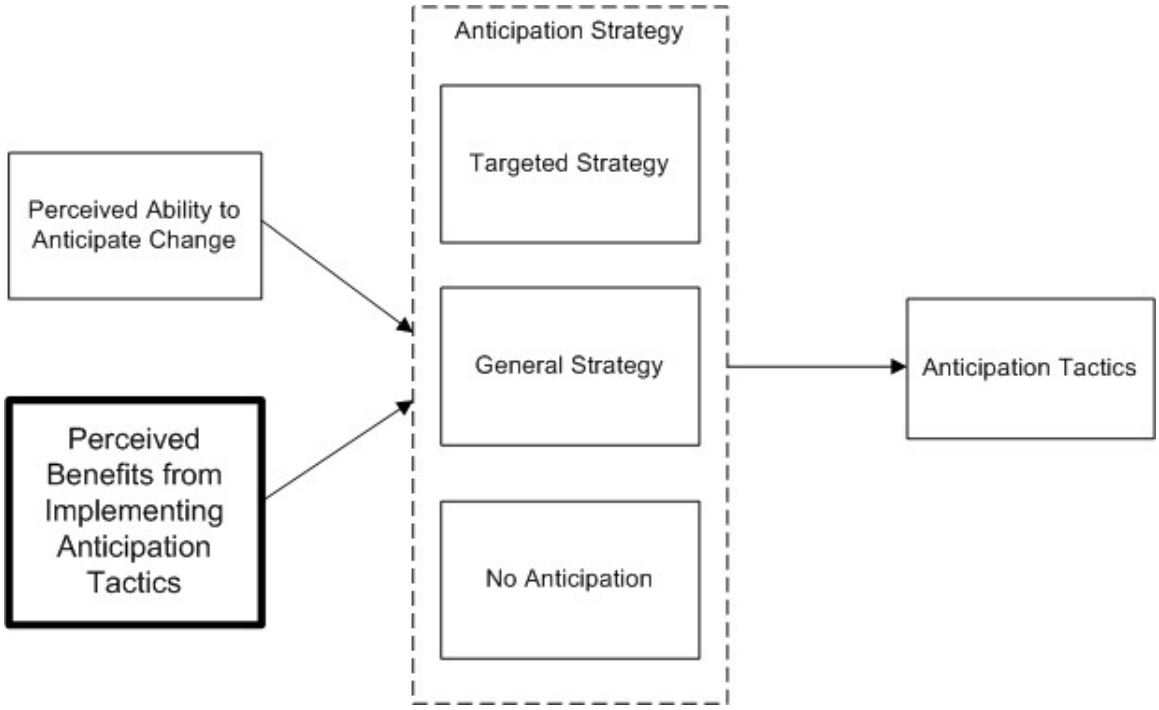

Figure 4-8: Perceived Benefits from Implementing Anticipation Tactics

A number of informants indicated that there were significant benefits to implementing anticipation tactics (Table 4-36). Anticipation of IS change was considered beneficial by informants discussing this concept (Table 4-37), even if anticipation never took place. In spite of the payoff of preparing for change, some informants acknowledged that time and resource constraints prevented anticipation efforts. This is an interesting finding given that some

informants mentioned that the additional time commitment to implement anticipation tactics was minimal at times.

Table 4-36: Interviews Indicating Perceived Benefits of Implementing Anticipation Tactics by Organization

	RetailCo (of 5)	InsuranceCo (of 5)	SoftCo (of 5)	PowerCo (of 5)	BankCo (of 1)	Total (of 21)	Percentage (of 21)
Perceived Benefits of Implementing Anticipation Tactics	2	3	1	2	1	9	43%

Table 4-37: Perceived Benefits of Implementing Anticipation Tactics - Illustrative Quotations

Perceived Benefits of Anticipation Tactics	Company	Informant
"The trade-off is typically…it may take you 40 extra hours now, but over the next 10 years that's going to save you hundreds and hundreds of hours. And that's the trade-off …a lot of teams with systems like ours spend the majority of their time doing maintenance. We spend the majority of time doing new development. And it's because we're saving all those hours on the front-end that allows that. And that's really your big trade-off. It takes a little longer to get the project done up front, but over time, you spend more time working on the new stuff rather than just maintaining."	InsuranceCo	IT Development Manager
"There's been a number of times where we anticipated a change up front in the design and somebody came back and said, 'you know what we need, we need to make it so that it does this instead.' How difficult is that? Well it's not difficult at all. We'll change the table. You know we don't even have to change the code." "I have seen that there's not really a significant increase in time in turning that code out in the first place. You've still got to test it all. You've still got to code it some way, whether you code it this way or whether you code it that way. Now, your coding may be faster if you just do it straight through and don't modularize it…you probably save a little bit of time there but it's not going to be a lot. Not compared to your maintainability of the program later."	InsuranceCo	Senior Developer B
"It's kind of like there's the right way to do things and then there's the way we're going to do it because our time is under pressure. In the long run does that cost us time? Absolutely, absolutely it does. If we could do it once and do it right that would be ideal but there's never time to do that it seems."	RetailCo	ETL Analyst

4.3.3.1 Implications of Perceived Benefits of Anticipation Tactics

As previously mentioned, informants agreed that implementing anticipation tactics were beneficial, but not all organizations actually follow through with anticipation efforts. For those

that did not follow through, there were commonly time and resource pressures requiring a solution in the short term. For organizations, this may present an opportunity to embrace and reward anticipation efforts by IS personnel. It is already recognized as beneficial by at least some individuals in IS departments, perhaps making it easier to incentivize.

4.3.4 Change Anticipation Tactics

The products of the change anticipation strategy are the change anticipation tactics (Figure 4-9), which are structures and features incorporated into the design of the system to decrease the difficulty of future changes to the application.

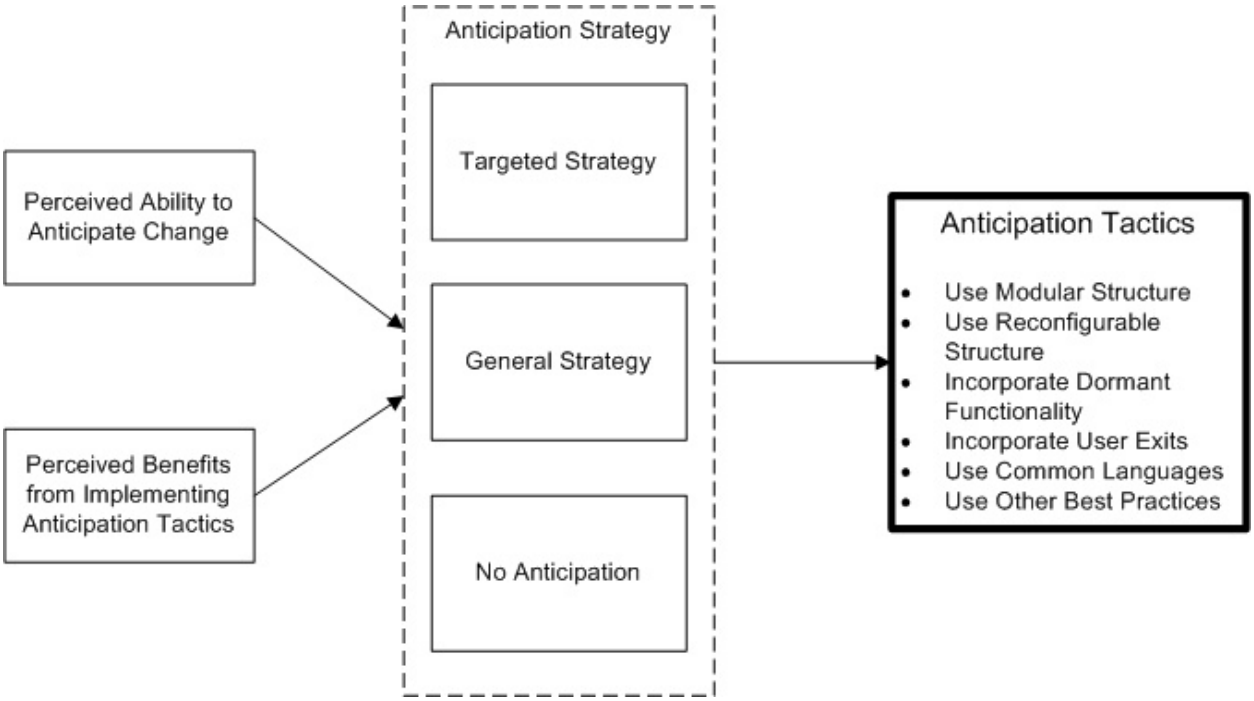

Figure 4-9: Anticipation Tactics

There are a variety of anticipation tactics that organizations are using to prepare for change, some of which were unexpected as the new anticipation model emerged. For example, modular structures and table-driven design (a type of reconfigurable structure) are commonly taught and utilized techniques for developers. Dormant functionality, however, was not an

expected finding, but is an interesting anticipation tactic because it is based on clear expectations of future changes to the application. In the following table, the working explanation for each of the tactics is shown (Table 4-38). While this list is not considered complete necessarily, it does cover the basic anticipation tactics discussed in the interviews.

Table 4-38: Explanation of Various Anticipation Tactics

Anticipation Tactic	Explanation
Use Modular Structure	The construction of software code into logical segments for organization and management purposes. This may be done at object level, service level, or other
Use Reconfigurable Structure	The construction of software code which abstracts variables, making those variables easily changed. Those variables can be changed to modify the operation or properties of the application. Typically, reconfigurable structures utilize database tables to specify variable values
Incorporate Dormant Functionality	The addition of functionality into an application which is not enabled, but may be at a later date
Incorporate User Exits	Places in software that allow for the opportunity to leave one application and interact with a second application, then return to the original application with perhaps some changed values
Use Common Languages	Standardizing so that applications are built with software languages that are designed to work together
Use Other Best Practices	Utilization of current best practices, such as object-oriented programming and technical standards

These anticipation tactics appeared in a number of interviews (Table 4-39). The most common tactics mentioned were reconfigurable structures and modular structures. Again, these techniques are well understood by many developers, though they may be underutilized given the amount of change applications undergo in current business. Other best practices were not mentioned, but this is not unexpected since these techniques are probably taken for granted, and consequently, minimized.

Table 4-39: Interviews Indicating Various Anticipation Tactics by Organization

Anticipation Tactic	RetailCo (of 5)	InsuranceCo (of 5)	SoftCo (of 5)	PowerCo (of 5)	BankCo (of 1)	Total (of 21)	Percentage (of 21)
Use Modular Structure	3	2	1	1	1	8	38%
Use Reconfigurable Structure	3	5	1		1	10	48%
Incorporate Dormant Functionality		2			1	3	14%
Incorporate User Exits		1	1			2	10%
Use Common Languages				1	1	2	10%
Use Other Best Practices						0	0%

In the following paragraphs, each of the anticipation tactics will be discussed individually. First, informants discussed their logic for utilization and construction of modular designs (Table 4-40). The general philosophy of modularity is to break the application into logical components for organization and management purposes. Modular structures are also intended to reduce complexity relative to non-modular structures, where many lines of software code are all grouped together and harder for developers to process.

Table 4-40: Anticipation Tactics - Illustrative Quotations of Modular Structure

Change Anticipation Tactics – Use Modular Structure	Company	Informant
"One of the things we try to do when we design a system is to layer it so that the presentation piece is separated from the business logic, which is separated from any calculations we might have to do, which is separated from any reports that we might have to run. The idea there being that most of the changes that we would be hit with…would probably affect one section of the program rather than the program as a whole, if they were more tightly interwoven. But because they're layered and because each layer has different responsibilities, it's easier to change that one little section and test it without having to test everything."	InsuranceCo	Senior Developer B

Reconfigurable structures are commonly used as anticipation tactics because it is an easier alternative to changing aspects of an application than altering code (Table 4-41). Reconfigurable designs commonly are based on table-driven designs. Table-driven designs,

which were clearly mentioned on a number of occasions, abstract variables from the code and place them into database tables, which are more easily changed than software code and does not require recompiling.

Table 4-41: Anticipation Tactics - Illustrative Quotations of Reconfigurable Structure

Change Anticipation Tactics – Use Reconfigurable Structure	Company	Informant
"If when you're building a program, you know that six months or a year from now, you're going to have to change the rating part of it. There may not really be any code changes as far as how it calculates the rate, but your data tables may change, your base rates that it uses. And so you try to write the program so that you can easily update your rating tables or your base rate tables without having to necessarily impact the calculation routine.	InsuranceCo	Senior Developer B
"We try to, when we write our systems, think ahead on those things to anticipate where our most frequent changes will be. 'Who's going to get what email?' We're not going to program that. We're going to make it table-driven."	InsuranceCo	Senior Developer A

Dormant Functionality is the addition of functionality to the application that is disabled initially, but included so that it can be enabled when it is needed (Table 4-42). Perhaps more so than any other anticipation tactic, dormant functionality clearly indicates a targeted strategy and definite expectations of how an anticipated change can be prepared for. At this point, dormant functionality seems to be an advanced capability, and is not expected to be as widespread as reconfigurable structures.

Table 4-42: Anticipation Tactics - Illustrative Quotations of Dormant Functionality

Change Anticipation Tactics – Incorporate Dormant Functionality	Company	Informant
"We have not implemented all of that target state because we haven't had the requirements to drive that. A good example of that would be our architecture has a component of it called guaranteed messaging, so we're talking between our system and another system. There are use cases that require guaranteed messaging. We've not implemented guaranteed messaging in two years because we have not had a requirement that has come through that has necessitated us implementing that functionality of our SOA stack." [but it's built in already]	BankCo	SOA Manager

While not mentioned frequently in the interviews, user exits are a common anticipation tactic utilized in applications. The extent to which user exits are implemented in custom development within organizations is unclear, but their utility as a preparation tool for IS change is certain. While alluded to in a number of instances, interviews clearly discussing user exits as an anticipation tactic were not present. For illustration purposes, a borderline example (possibly considered as an IS change example or an anticipation of IS change example) is included in Table 4-43.

Table 4-43: Anticipation Tactics - Illustrative Quotations of User Exits

Change Anticipation Tactics – Incorporate User Exits	Company	Informant
"We already have a place where an adjustor goes in, sees his work list for the day, and he prints a PDF of all of those documents so that he can just hit print, grab all of his first notes of loss and stuff, and go. On the end of that document we're going to take those addresses and we haven't decided on the service but utilize a third party service to do the route planning and send that back to us and display a map with, you know, driving directions - equal to a Map Quest type functionality at the bottom of the PDF."	InsuranceCo	IT Development Manager

Utilization of common languages also serves as an anticipation tactic (Table 4-44), though in a more basic sense than previous examples. Common languages can help prepare for changes in both a technical and a knowledge sense. In a technical sense, anticipation of future IS change needs may indicate that connections with other applications will be necessary, and choosing the appropriate languages at the start will facilitate those connections. In a knowledge sense, by utilizing common languages, personnel can more easily transfer their skills with those languages to other projects and "ramp up" more easily.

Table 4-44: Anticipation Tactics - Illustrative Quotations of Common Languages

Change Anticipation Tactics – Use Common Languages	Company	Informant
"I will tell you one decision we made, and I told you we are a Microsoft shop. I think in 2001, we had a…change and we said we are going to do C# and that's all we're going to code in. [We're doing this for] a couple of reasons. C# is as close to Java as I could get so I am hedging my bets. Because I realized a lot of systems that we buy on the external market are Java based, and I will eventually have to support those, or do some kind of connection with them. That one decision has really helped … Everyone has a common language now that they can communicate in, all of our developers. We found we were organized in a portfolio model, we have a team that supports marketing's applications, a team that supports accounting's side of the house. And used to, those were pretty isolated islands. Today in crisis, I can move someone from one portfolio to another. Although they may not know the business logic, they can look at the code, and say I understand the model, I understand the language that it's written in, if I have to solve a business problem, I can do that. We've seen a lot of benefits to that standardized kind of stuff that I didn't think we we're going to make. I was really trying to hedge my Java bet at the time, but it turned out to have a lot of downstream benefits that I didn't think of at the time."	PowerCo	IT Architect

Again, no informants mentioned other best practices in a context that could be interpreted as an anticipation tactic (Table 4-45). However, it is expected that utilizing best practices, such as object-oriented programming and technical standards (e.g. XML), organizations will be better prepared for IS changes. Essentially, other best practices is a default position for many IS departments at this point. For completeness, other best practices is included as another anticipation tactic in that some IS changes will be more easily accommodated, even if "by accident".

Table 4-45: Anticipation Tactics - Illustrative Quotations of Other Best Practices

Change Anticipation Tactics – Use Other Best Practices	Company	Informant
No informant mentioned Other Best Practices as an anticipation tactic in the interviews.		

4.3.4.1 Relationship between Anticipation Strategies and Tactics

Based on the change anticipation model and the evidence presented, a logical next step is to link particular anticipation strategies with particular anticipation tactics. At this time,

evidence of the relationships is only sufficient for speculation. While certainly an avenue for more research, it is believed that targeted anticipation strategies lead to particular anticipation tactics, for example. There is overlap though, and the differences seem to be delineated based on the mindset associated with each strategy (Table 4-46).

For example, modular structures can be designed and implemented from either a targeted perspective or a general perspective. In a targeted case, modules can be built around particular expected business or technical changes. In other words, modules can be structured to accommodate an expected change within a single module, or between modules (where internal module workings are untouched), based on whichever is deemed most desirable. In the general case, modules can be developed without predicting specific changes. However, it can still follow a logical structure, and may accommodate a variety of IS changes more easily than non-modular designs, even if particular changes are unknown.

Reconfigurable designs also fit both targeted and general strategies. The same logic applies. If particular changes are anticipated, then tables or reconfiguration files can be constructed in preparation for those changes. If particular changes are not, or cannot, be anticipated, developers may still take opportunities where identified to abstract variables into tables as a hedge against future IS changes. User exits may also be designed from both perspectives.

Dormant functionality seems to apply only to targeted strategies. This tactic requires identification of a particular functionality during design that is included in the application, but disabled. Common languages and other best practices seem to apply to general strategies because they are in preparation of change, but not predicting particular changes.

Table 4-46: Potential Relationships between Anticipation Strategies and Anticipation Tactics

Change Anticipation Tactic	Targeted Strategy	General Strategy
Use Modular Structures	✍	✍
Use Reconfigurable Structures	✍	✍
Incorporate Dormant Functionality	✍	
Incorporate User Exits	✍	✍
Use Common Languages		✍
Use Other Basic Best Practices		✍

The anticipation tactics shown here are becoming more common in packaged software, but it is not clear at this point how often these tactics are a part of custom developed software. More investigation may reveal that packaged software depends more on these tactics because of its diverse users, and therefore, are more conscious of preparing for a wider variety of changes. Nonetheless, the anticipation tactics incorporated into packaged software still affect the organization's future ability to change the application easily, and therefore, should be considered during acquisition.

4.3.4.2 Implications of Change Anticipation Tactics

The change anticipation tactics from the anticipation stage (development) become the application characteristics relevant to change during the response stage (post-implementation) (Figure 4-10). This is the link between the IS Response Model (Figure 4-1) and the Change Anticipation Model (Figure 4-5). Effectually, if the application is not prepared for particular IS changes during the anticipation process by incorporating anticipation tactics, then when that particular IS change occurs after deployment, the application will be less able to accommodate that change easily. While reconfiguration is the most desirable IS response, at times, a well-planned system should be more easily recoded or expanded due to existence of anticipation tactics, such as modular design or user exits.

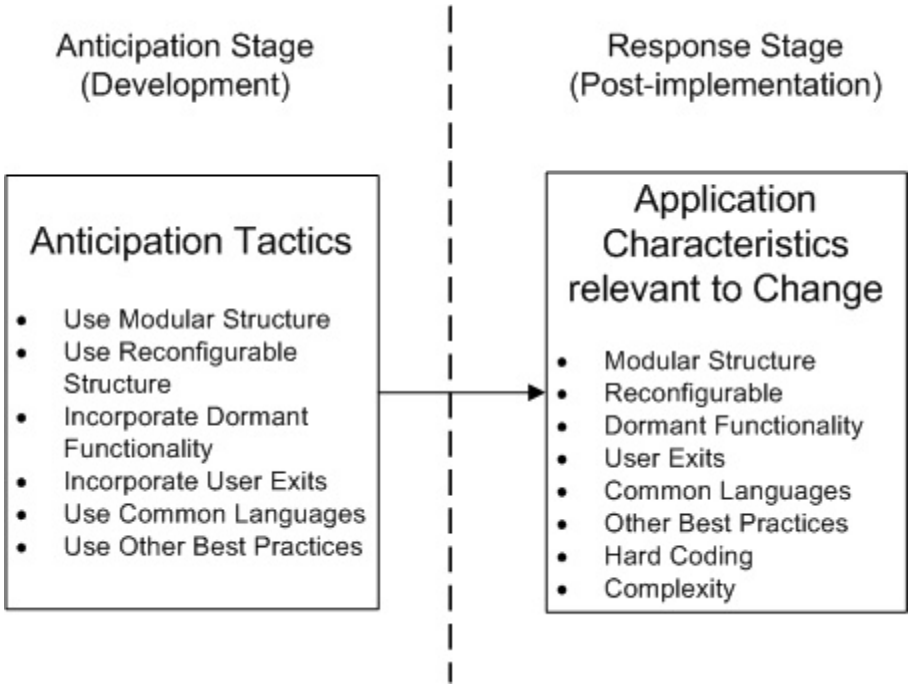

Figure 4-10: Link between Anticipation Tactics and Application Characteristics Relevant to Change

One implication of change anticipation tactics is that there are a variety of different tools available to developers to prepare for future IS changes. With the link between anticipation tactics and application characteristics relevant to change, developers have a stronger argument to support their perceptions of the benefits of implementing anticipation tactics.

Although most of the anticipation tactics have an analog in application characteristics relevant to change, the latter list also includes complexity and hard coding. These are not necessarily desirable characteristics during the response stage and may originate as byproducts of choices made during development of the application. In other words, complexity may arise partly due to the structure that developers choose.

An additional point is that the application characteristics relevant to change includes some additional components which are not included in the anticipation tactics. This is because

there may be some undesirable byproducts of designs, such as complexity, which affect the IS response but are unintended, and thus, not considered anticipation tactics.

CHAPTER 5 CONCLUSIONS

"We are ready for any unforeseen events that may or may not occur." –Dan Quayle (Chalfant 1990)

5.1 INTRODUCTION

This chapter begins with a synthesis of the various models and concepts developed and discovered during this study. The goal of the model synthesis is to connect the initial conceptual model, which was developed in advance of data collection and used to guide this research, with the anticipation model developed during the study. Next, a brief summary of the findings related to the research questions is given. In addition, a summary of findings related to the anticipation model is also included. Finally, limitations of the study, contributions for theory and practice, implications of the findings, and future research directions are discussed.

5.2 MODEL SYNTHESIS

This research study began with an interest in understanding the business-driven IS changes required in organizations, the application characteristics relevant to those changes, and the IS responses that were available given that combination. The relationships between these ideas were the basis for the initial conceptual model guiding this research (Figure 5-1).

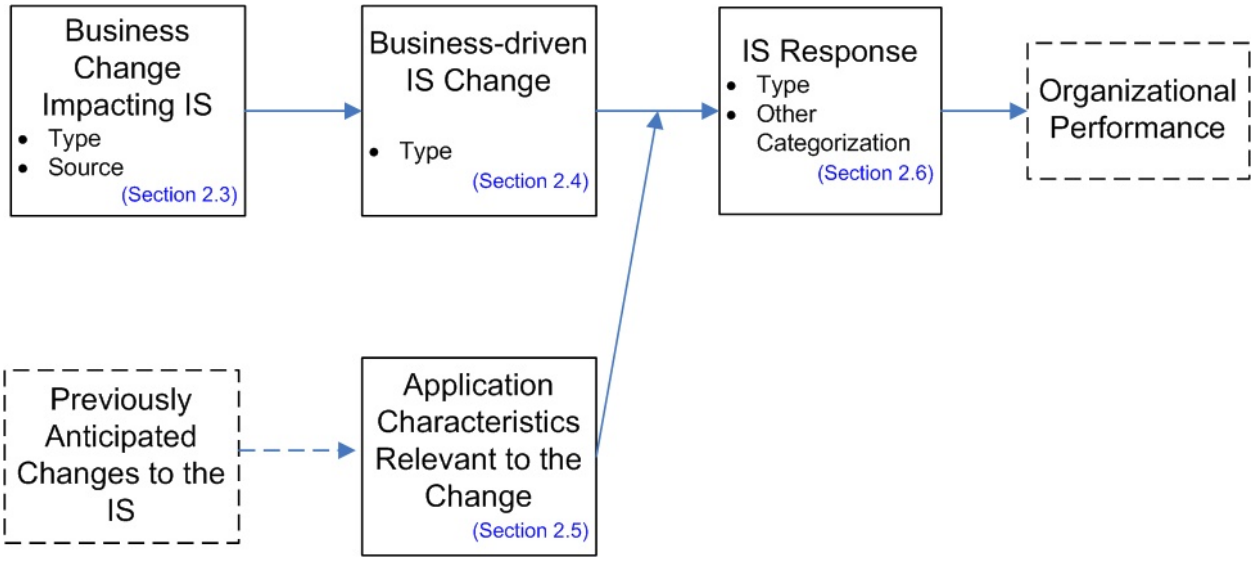

Figure 5-1: Initial Conceptual Model (revisited)

The business-driven IS changes are tied to the organizational need for change because, at times, organizational need for change requires changes to the organization's business processes, and in turn, to the organization's information systems. Part of the significance of this research to the IS field is based on the premise that organizational agility is connected in some situations to the ability to change information systems, or the IS Response Ability. The relationship between organizational agility and IS response ability was depicted in the Bi-directional View model (Figure 5-2) developed in Chapter 2.

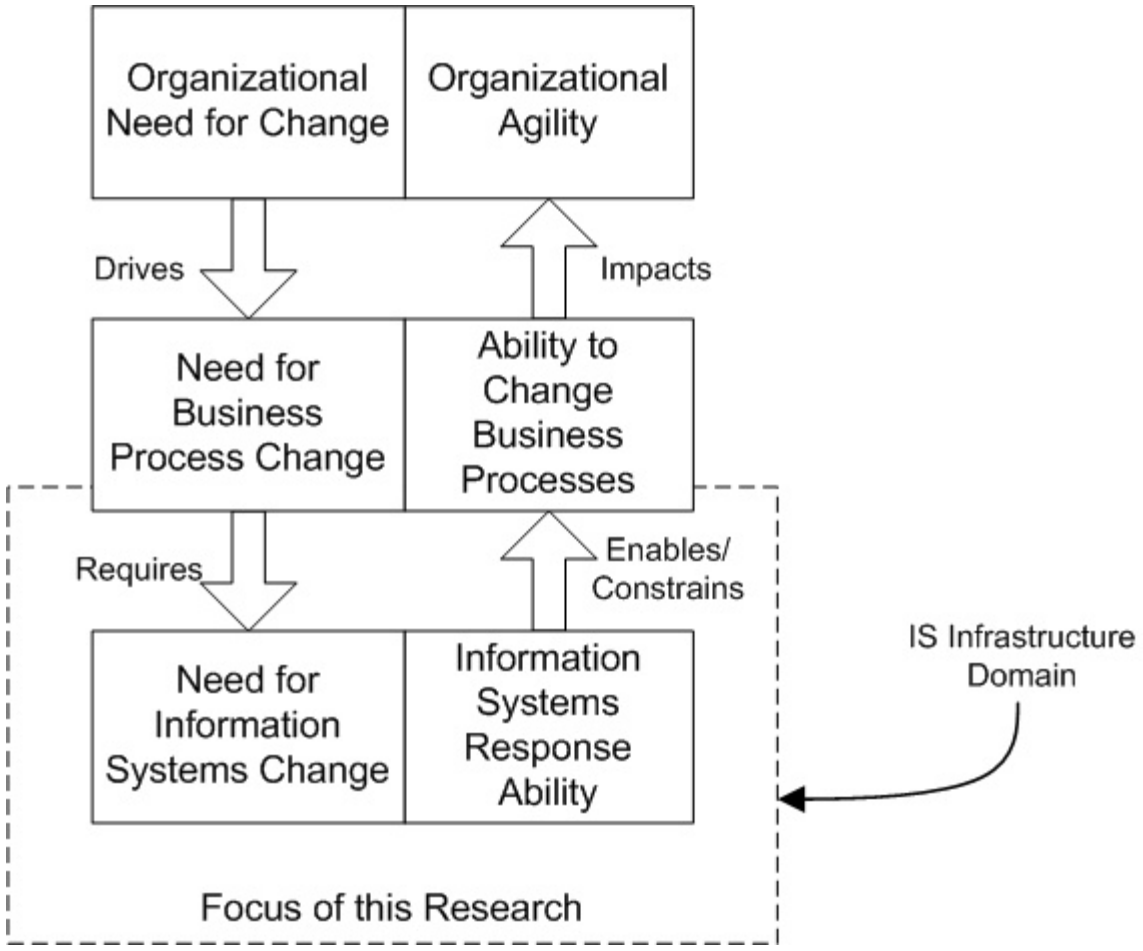

Figure 5-2: Bi-directional View of the Relationship between Organizational Agility and IS Response Ability (revisited)

As data collection efforts progressed, it became apparent that a concept from the initial conceptual model, "Previously Anticipated Changes to the IS", was a component of discussions by the informants. The interviews were adjusted to include more exploration into concepts surrounding anticipation of IS change and resulted in the eventual IS Change Anticipation Model (Figure 5-3). This model incorporated discussions about perceptions of the ability to anticipate change, perceptions of the benefits of implementing anticipation tactics, various anticipation strategies, and various anticipation tactics.

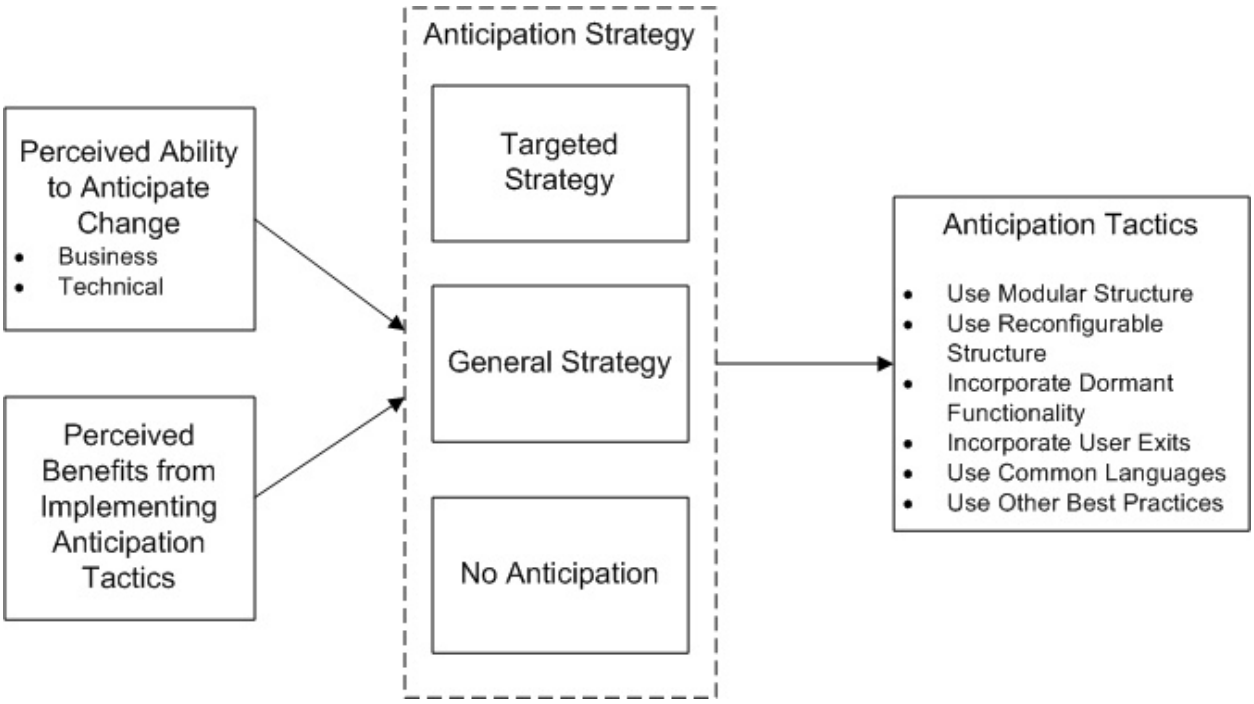

Figure 5-3: IS Change Anticipation Model (revisited)

Subsequently, a link was drawn between Anticipation Tactics during the anticipation stage and Application Characteristics Relevant to Change in the post-implementation stage (Figure 5-4). This connection solidifies the assertion that IS responses are, at least in part, dictated by choices made during design rather than choices made when IS change is necessary.

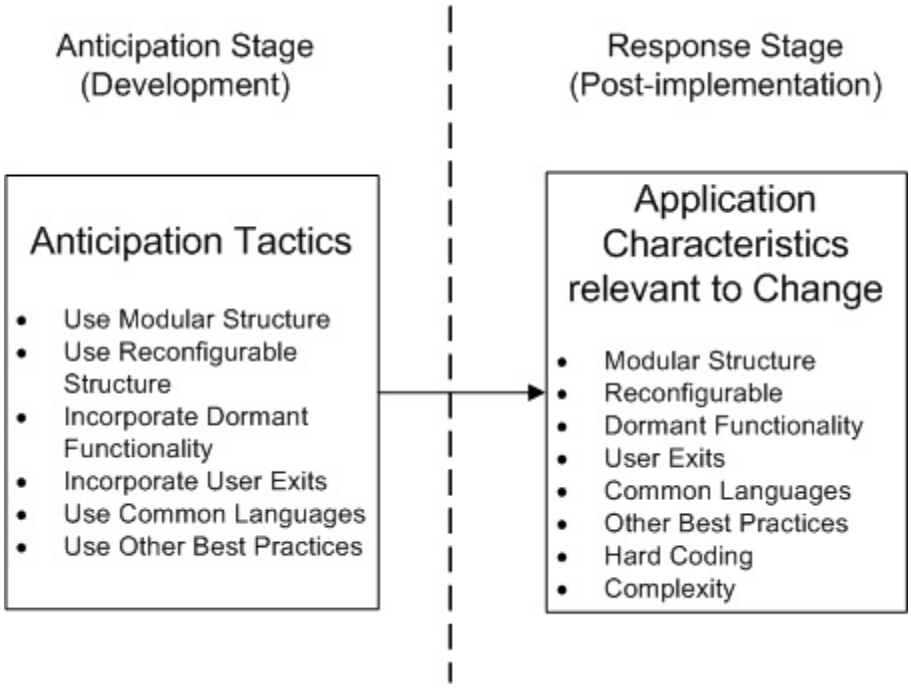

Figure 5-4: Link between Anticipation Tactics and Application Characteristics Relevant to Change (revisited)

Each of these models serve to paint a picture around the subject of Business-driven IS Change. While verbal descriptions have been included along the way to indicate connections between various ideas and models, for clarity, the models and concepts have been assembled into the following Synthesized Model (Figure 5-5)

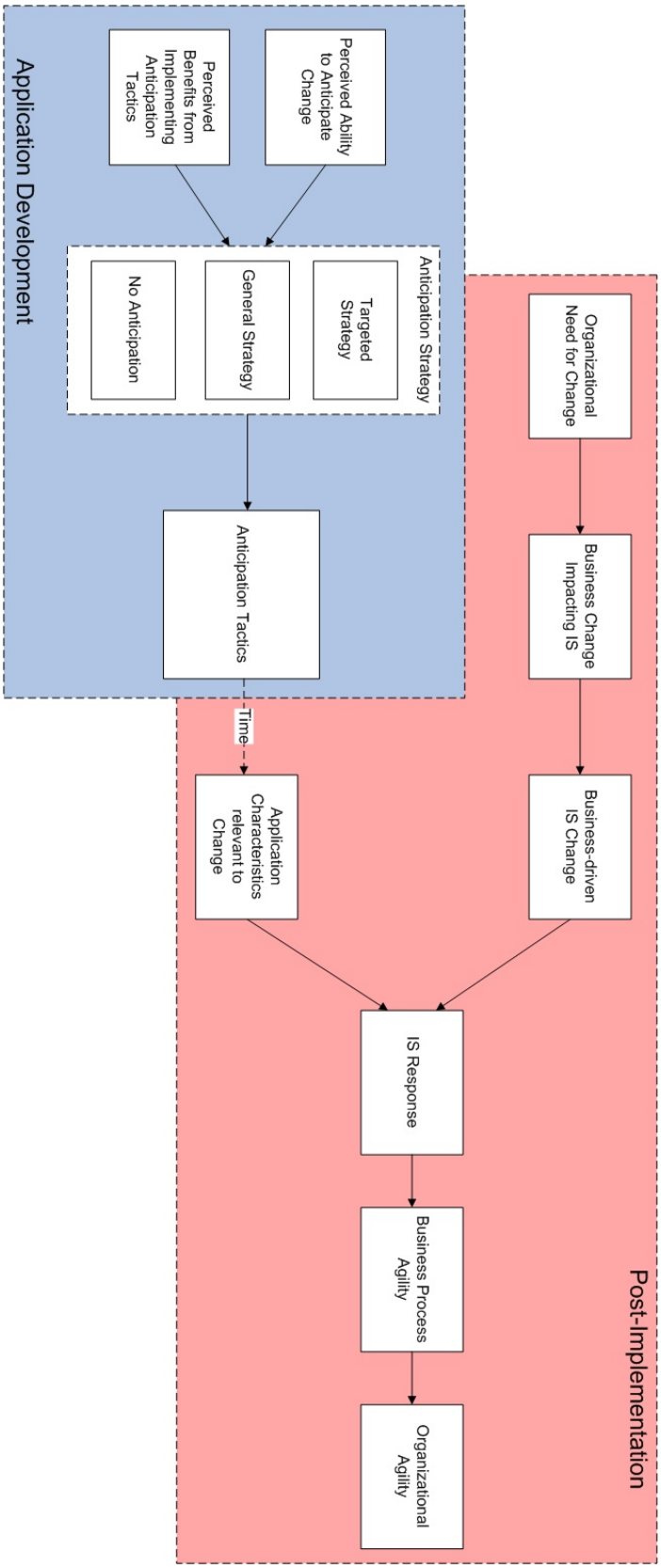

Figure 5-5: Synthesized Model of Business-driven IS Change

5.3 SUMMARY OF FINDINGS

This study began with three primary research questions, but was open to findings beyond those initial research questions. Using the synthesized model (Figure 5-5) as a basis for discussion, this section will review the findings of this study. The three primary research questions are related to Business-driven IS Change, Application Characteristics relevant to Change, and IS Response, all of which are constructs depicted in the Post-implementation section of the synthesized model. Additional findings are related to the Application Development section of the synthesized model.

5.3.1 Findings Related to the Initial Research Questions

This research began with three primary research questions. Twenty-seven semi-structured interviews with thirty-one individuals were conducted in order to find some answers to those research questions. Of the twenty-seven interviews, twenty-one were coded using an iteratively developed coding template. Six interviews were leveraged using field notes, but were not included in the coding process because they were not recorded. Following are the findings related to the original research questions.

The first research question for this study was: *What are the types of change required of the IS by the business?* From the interviews, we were able to find a few different angles to view business-driven IS change. First, a typology of changes was created using examples from the interviews. These changes included data integration changes, user interface changes, communications changes, reporting changes, business process changes, new functionality changes, and data collection changes. These changes have been generalized in order to avoid implying a particular application infrastructure. In other words, data integration changes may happen to legacy architectures, enterprise resource planning systems, service-oriented

architectures, or other architectures. These abstracted business-driven IS change types may serve as a starting point for evaluating the ability of the various architectures to respond to the required types of changes.

Second, business-driven IS changes were labeled, where possible, as anticipated or unanticipated. In general, those changes that were considered anticipated were more easily accommodated than those that were unanticipated.

Third, business-driven IS changes were labeled, where possible, as local or interdependency changes[11]. Local changes were changes that occurred within an application or module, but did not affect the relationship with other applications or modules. In general, local changes are easier to deal with in terms of managing complexity because changes do not ripple through other systems. Additionally, testing is typically less extensive in local changes. Interdependency changes are changes that affect the relationship between applications or modules. When the interface between applications or modules change, it often results in more complexity for the IS personnel to deal with, and significantly more testing.

The second research question for this study was: *What are the application characteristics that impact the ability to change?* Analysis of the interviews indicated a number of application characteristics that were relevant in particular change situations. Some of these application characteristics positively impacted the ability to accommodate a change need, while others negatively impacted the ability to respond. Among the positive application characteristics are modular design, reconfigurable design, dormant functionality, user exits, common languages, and hidden complexity. The negative application characteristics were hard coding, complexity that is not hidden, and interdependency complexity.

[11] It should be noted that previous design decisions play a role in whether a change is local or interdependency-related.

The third research question for this study was: *What are the response options of the IS, given a particular IS change need and a particular software infrastructure?* This study was able to partially answer this research question. Change examples that grouped particular IS change requests with particular application characteristics, which yielded particular IS responses were not available with enough consistency to be fully confident in the analysis. Due to the inconsistency between change examples (described in greater detail in the following limitations section), the analysis of the interview data was reported at the interview level rather than the change level. However, the data did support that there were three primary IS responses, reconfiguration (Type I), recoding (Type IIA), and expansion (Type IIB). A variation of recoding responses was also found. At times, developed choose a recode and restructure response in order to meet current change needs, and also prepare for future changes.

The findings related to the research questions help fill a few gaps in existing research. First, this research extends IT flexibility research, which discusses changes that fall within a set of anticipated IS changes, by considering a broader range of IS changes, some of which are unanticipated and fall beyond the flexibility of the applications. In particular, this research describes IS responses, such as recoding and expansion, for cases where reconfiguration is not enough to meet the change need. That IS changes can be beyond the range of changes anticipated and factored into the application's design shifts the focus from flexibility as a property of a system to flexibility as an interaction between the particular IS change needed the application characteristics relevant to that change. In other words, no matter how flexible a system is believed to be, when a required change falls outside of the range of reconfiguration options, other IS responses, such as recoding or expansion are necessary. Even then, the

difficulty of making the change is affected by the other characteristics of the applications (e.g. modularization, etc.)

The second gap addressed by this research is an alternate link between IS and organizational agility. Previous research has focused on the capabilities created or provided by IS which, in turn, affect decision making for the organization. In this way, IS helps organization sense and respond to threats and opportunities, which potentially makes the organization more agile. Following initial work by Chen (2004), this research looks at the case where the organizational need for change requires a change to the organization's information systems. Accordingly, the ability to change the organization's IS, or the IS response ability, affects the ability of the organization to be agile. This view is supported in part by previous research which indicates that organizations often struggle to change their IS. The extension made by this research is that the IS response ability is affected by both the particular IS change need and the application characteristics relevant to that change, the latter of which, can be affected by anticipation of IS changes during application design.

At the outset of this research, it was unclear whether the initial conceptual model (Figure 5-1) would be the model shared by the informants. With respect to the initial conceptual model, it seems that the informants did perceive the major constructs, if not the model itself. Informants did seem to have a view of how business-driven IS changes interact with particular application characteristics to make change easy or difficult. The model takes the additional step of formalizing categories and types that may not be formulated in the minds of practitioners.

5.3.2 Findings Related to Anticipation of Change

While "Previously Anticipated Changes to the IS" was a construct in the initial conceptual model developed in Chapter 2, it was not within the scope of the original study. This construct

was included in the model to indicate recognition that anticipation of changes during the development phase could impact the application characteristics, which could in turn, affect the IS response. In other words, decisions about using modular design, for example, made during development could have impacts years later when IS change is necessary. This concept was an important insight because of its potential effect on IS responses, but at the start of the study, it was outside of this study's interests.

As interviews were conducted though, it became more apparent that anticipation of change during development was an important avenue for expansion and should be included as a component of this study. As a consequence, interview questions were expanded to ask more specifically about anticipation efforts, if they were not discussed by the informant without prompt. Interview data was analyzed with anticipation in mind. This led to the creation of the Change Anticipation Model (Figure 5-3), which is based on emergent findings from the interviews centered on four major constructs.

The first finding related to change anticipation is that a number of informants indicated that anticipating change is beneficial to the organization because it can save valuable time and resources when IS change is necessary after implementation of the application. The acknowledgement that there are perceived benefits to implementing anticipation tactics not only indicates that change anticipation is within the purview of some informants, but that it is also important.

The second finding related to change anticipation is that some informants perceived that they have different abilities to perceive change. In fact, there are differences between informants on whether they can effectively anticipate changes from a business perspective or a technical perspective. In other words, some informants may have enough experience with the business to

anticipate some changes that they should prepare for. Others may have insights into common technical changes, and prepare for those accordingly.

The third finding is that there are three different strategies, or "mindsets", for anticipating change. First, some informants indicated that they took a targeted approach to anticipating changes, from either a business or technical perspective (or both). Targeted strategies are concerned with predicting specific changes and preparing for them. Second, some informants indicated that they took a general approach. General approaches do not try to predict specific business or technical changes. Instead, a general approach looks for aspects of the application that can be made flexible in preparation for future change. Finally, some informants indicated that they had "no anticipation" strategy. This does not necessarily indicate a belief that it is unimportant to anticipate change, but that there may be limited time and resources which preclude anticipation efforts.

The fourth finding is that there are different anticipation tactics utilized by developers to prepare for change. Modular designs, reconfigurable designs, dormant functionality, user exits, common languages, and other best practices are all examples of tactics that are incorporated into applications to prepare for change. Furthermore, these tactics can fit one or both of the proactive anticipation strategies (targeted or general).

The fifth finding related to change anticipation is that it did not appear to be a formal process for the organizations in this study. The most consistent views of anticipation came from InsuranceCo, but it was not clear that it was more than the culture of the IS department. In other words, it appears at this point that anticipation efforts within the various organizations are individually initiated and guided.

At this point, we do not believe that the practitioners we spoke with have a mental model of change anticipation as developed and formalized in this study. Some do seem to clearly understand the potential benefits of anticipating change and that there are tactics that can be incorporated to prepare applications for change. However, they do not appear in most cases to have a formal approach to this process. Future research should seek to validate this model with practitioners.

5.4 RESEARCH LIMITATIONS

This qualitative research study used semi-structured interviews with practitioners from IT departments in a small collection of organizations. While this method is useful for uncovering a great level of detail surrounding a phenomenon, it also has weaknesses. These weaknesses should be considered when interpreting the findings of this study.

The first research limitation is related to the lack of consistency around IS change events. The limitation required the unit of analysis to be elevated to the interview level. This eliminated detailed examination of the interaction between a particular business-driven IS change and application characteristics relevant to that change, and the resulting IS response. This partially impacted the ability of the study to answer research question three. In other words, this study was able to examine the IS responses generally, but not as an interaction between particular IS changes and particular application characteristics. Even if we had examined the interview data at the change level, it is unlikely that a predictive model could have been produced to fit all combinations of IS changes and application characteristics.

The sample for this research, while theoretically appropriate and representative of typical issues in large organizations, was selected partly on convenience. Personal contacts for the primary researcher served as initial contact points for the IT departments in the organizations,

and subsequent interviews were acquired through successful initial visits. Again, while the organizations represent typical organizations, it is important to disclose that the sample was convenience-based.

The individuals interviewed for this study represented a variety of roles within the IT departments. By interviewing a range of individuals, this research benefited from many different perspectives on the issues surrounding business-driven IS change. An alternative design could have spoken to individuals in a single role within the organizations and garnered more detail about their perspectives, but given the infancy of this research area, the ability to look at the issues from several different vantage points was deemed preferable.

The interviews were conducted by the primary researcher and analyzed by both the primary researcher and advisor (in part). Therefore, the biases and interpretations of the researcher could influence the findings reported in this document. The research base for this research area is not well developed, so there is less opportunity to compare the concepts developed here with previous findings. As with any youthful research, additional research will serve to help validate or disprove the concepts and models introduced here.

5.5 RESEARCH CONTRIBUTIONS TO THEORY

While previous research has indicated that information systems play a role in organizational agility (Sambamurthy et al. 2003), for the most part, the context in that research is related to sensing opportunities or problems and responding at the organizational level. That is, gathering and providing information and knowledge for better decision making. This research looks at an alternate situation where the organization's information systems are required to change to meet agility needs, such as changing business processes to "respond" to recognized opportunities or problems. As a result, the ability to change information systems affects the

agility of the organization. This bi-directional view (Figure 5-2) of the relationship between organizational agility and the ability to change the information systems, or IS Response Ability, is the first contribution to theory.

Second, this research sharpens the focus on business-driven IS change by realizing that the ability to respond to IS change needs is dependant both on the particular IS change and application characteristics relevant to that change. On the surface, this argument appears obvious. However, previous research simply acknowledges the impacts of existing infrastructure on the ability to change (Overby et al. 2006), but does not look in detail at what those characteristics of the applications are.

Third, this research lays the groundwork for exploring the types of Business-driven IS change. The various types of Business-driven IS change (e.g. data collection, process change, new functionality, etc.) are independent of particular application types and should provide some future research opportunities for evaluating which types of changes are best accommodated by which architectures.

As a surprise resulting from the exploratory nature of this research, the role of anticipation of change in both application design and accommodation of future IS change has become a key theoretical contribution. While previously anticipated IS change was acknowledged in the initial conceptual model, much more detail emerged around this idea from the interviews, resulting in a new IS Change Anticipation Model.

Finally, the IS Change Anticipation Model has created a framework of anticipation strategies and anticipation tactics that are employed at various organizations. This model will guide research going forward in anticipation research, and the relationship between anticipation in system development and the ability to ultimately meet agility needs at the organizational level.

5.6 RESEARCH CONTRIBUTIONS TO PRACTICE

Managers at both the organizational level and in IT departments are frequently struggling with the need to be agile. For top managers, this research develops the rationale behind the bi-directional relationship between organizational agility and the organization's information systems. This is important because of the popular view of the relationship between agility and IT is rooted in providing better and faster information for decision making. However, many of the changes enacted at the organization level require changes in the processes of the organization, and consequently, the information systems. Again, this research attempts to balance the view of IS in achieving organizational agility.

For IT managers, the IS Response model provides a framework to view both the role of particular IS changes and the role of application characteristics relevant to the change in responding to the need for change. Initially, this provides a tool for helping the organization understand the difficulty of accommodating unanticipated changes to the applications. This is important because of the common mismatch between the apparent simplicity of a particular IS change to the requestor and the difficulty in responding to the change request due to the application characteristics relevant to that change.

IT managers and developers should also re-emphasize the importance of detailed analysis and design, not only for the present but also the future. This emphasis reflects the shifting trend in software: software once was designed to work, now it must be designed to change. This realization should lead IT managers to evaluate their anticipation strategies for application development. Developers and managers in this study indicated that the additional effort to implement anticipation tactics into applications was marginal, especially compared with the payoff when change is necessary. This assumes that the business changes anticipated and the IS

changes prepared for are often the ones that occur. The accuracy of this assumption needs to be tested and may depend upon contextual factors.

While it is apparent from the informants that either targeted or general anticipation strategies, and likely both, are a good idea in application design, it should be noted that this can be taken to the extreme, where far too much money and resources are spent building in capabilities that will never be used. Therefore, IT managers must be involved in predicting the likelihood of changes. IT managers need to become more involved in understanding the direction of the company so that it can be factored into anticipation decisions. Additionally, it is the responsibility of the IT manager to understand the environment of the system, both internal and external to the organization, so that volatility associated with change requirements can be anticipated. For some applications, the environment will be stable, and basic anticipation efforts may be sufficient. For other applications, the environment may be constantly changing because of any number of reasons, from uncertainty in managerial decision making, shifting marketing plans, or ultra-competitive marketplaces. These applications may require extensive anticipation efforts to make sure that the application does not inhibit future agility needs of the organization. In extreme cases of environmental uncertainty, targeted strategies may not be effective because predicting particular changes may be impossible. In this case, general strategies may be the best option.

Developers should realize that they have a number of tools to deal with the prospect of change, whether the changes are easily anticipatable or not. Where possible, developers can target particular changes and build in dormant functionality or build modules around particular business processes. When the future is less certain, developers can still create table-driven designs, for example, to be as prepared as possible.

Once top managers realize the link between anticipating IS change and future change performance, they have a clear incentive to involve IT department leadership in strategic discussions so that the IT department can have a better ability to anticipate change. Furthermore, through involvement in high level discussions about the direction of the company, IT departments will be better able to anticipate what the business will need in the future. After all, anticipation is an effort in predicting the future and information is key to improving that ability.

5.7 IMPLICATIONS AND FUTURE RESEARCH DIRECTIONS

It is perhaps the nature of exploratory research to create more questions than answers, and this particular study is no exception. On the surface, this research has sought to investigate and confirm an initial conceptual model and answer related research questions. For the most part, this seems to have been successful. However successful from an exploratory perspective it seems, there is plenty of work to do in a confirmatory sense.

This research took a broad perspective of business-driven IS change from many roles in several companies. While this was certainly beneficial for determining if individuals with different responsibilities think about IS change in the same way, it sacrificed consistency in the examples given. Future research could better isolate the individuals in various roles and take a more focused and stringent approach to interviews. This would enhance the examples of IS change by narrowing discussion to just a few specific examples rather than allowing informants to travel back and forth between concrete and hypothetical circumstances.

This research could also provide a way to examine whether some architectures handle IS change better than others. For example, it might be expected that service-oriented architectures are better equipped to handle business process changes than other architectures. By identifying IS changes without implying a particular architecture, the characteristics of the architectures can

be evaluated for comparison. The implication is that organizations that frequently change business processes might leverage one architecture better than others when change is necessary.

More research is needed to evaluate whether there are cases or conditions where it is preferable to wait until IS change is necessary to respond rather than preparing in advance. Informants discussed the tradeoffs of spending additional time up front during development to prepare for potential IS changes, but it is unclear whether the additional time and resources are worth it in all cases. For instance, companies with very stable environments and systems may be better served by a "wait and see" approach to dealing with IS changes.

While some additional questions regarding IS change anticipation are described in the previous chapter, several are worth revisiting here. Future research should also extensively explore the realm of anticipating IS changes. This research has conceptualized the IS Change Anticipation model with four primary constructs. This model is not concrete at this point. This model does seem to lend itself to quantitative measurement better than the IS Response model, which requires examples with context. Future research should look at developing an instrument to measure the anticipation strategies and tactics taken by IT departments.

In addition to an instrument, additional research needs to evaluate where anticipation strategies originate. There is some initial evidence that anticipation efforts start from the bottom-up with developers since they are the ones that must struggle with change at the technical level. In another example, it seems that anticipation efforts are encouraged, if not guided, by IT department management. In either case, it is not clear that the anticipation of IS change is as prominent as it could be during system analysis and development.

More research needs to be devoted to understanding how emerging technologies and development philosophies related to service-oriented architectures (SOA) and business process

management (BPM) fit in with this research. While the models created within this research are intended to be applicable to SOA and BPM, much of the interview data was based in more traditional application development environments. An underlying current of BPM is structuring and modularizing in a way that allows for easier change. This emphasis on preparing for change is not necessarily a new idea, but it is certainly more prevalent in SOA and BPM. For the foreseeable future, there will be questions about how the best way to divide services into modules to prepare for change. For organizations utilizing SOA and BPM, architects and developers will have to make conscious decisions how to prepare their systems for change.

It is also unclear which anticipation strategies are most effective. While targeted strategies prepare applications for specific IS changes, some informants have stated how difficult it can be to predict particular business changes. In fact, one study shows that as many as two-thirds of IS changes cannot be anticipated (Goodhue et al. 2008). If this is the case, then IS departments should weigh the payoffs of targeted strategies. The implication is that general anticipation strategies may be more effective for organizations. More research is needed to assess which of these strategies are most effective. It is expected that for most organizations a combination of targeted and general strategies will be most effective.

Finally, while the anticipation of change has been described as valuable when change is encountered later, this research has not explored the actual result, the eventual benefits, be it local efficiency of the IT department or overall agility of the organization, of particular change anticipation strategies and change anticipation tactics when IS changes are needed and made. Is more benefit derived from particular strategies or tactics compared to others? Is it truly beneficial to the organization to try to anticipate changes to their applications? In the end, these benefits are primary goals of change anticipation and should be examined.

REFERENCES

Ackroyd, S., and Fleetwood, S. "Realism in Contemporary Organisation and Management Studies," in: *Realist Perspectives on Management and Organisations*, S. Ackroyd and S. Fleetwood (eds.), Routledge, London, England, 2000.

Allen, B.R. "An Unmanaged Computer System Can Stop You Dead," *Harvard Business Review* (60:6) 1982, pp 77-87.

Allen, B.R., and Boynton, A.C. "Information Architecture: In Search of Efficient Flexibility," *MIS Quarterly* (15:4) 1991, pp 435-445.

Baldwin, C.Y., and Clark, K.B. *Design Rules* The MIT Press, Cambridge, MA, 2000.

Banker, R.D., Davis, G.B., and Slaughter, S.A. "Software Development Practices, Software Complexity, and Software Maintenance Performance: A Field Study," *Management Science* (44:4) 1998, pp 433-450.

Barua, A., and Kriebel, C.H.M., Tridas "Information Technologies and Business Value: An Analytic and Empirical Investigation," *Information Systems Research* (6:1) 1995, pp 3-23.

Bostrom, R.P., and Heinen, J.S. "MIS Problems and Failures: A Socio-technical Perspective - Part I: The Causes," *MIS Quarterly* (1:3) 1978, pp 17-32.

Boudreau, M.-C. "Using Grounded Theory in IS Research," Academy of Management Conference, Denver, CO, 2002.

Boyatzis, R.E. *Transforming Qualitative Information* Sage Publications, Thousand Oaks, CA, 1998.

Broadbent, M., and Peter, W. "The Implications of Information Technology Infrastructure for Business Process Redesign," *MIS Quarterly* (23:2) 1999, pp 159-182.

Browne, J., Dubois, D., Rathmill, K., Sethi, S.P., and Stecke, K.E. "Classification of Flexible Manufacturing Systems," *The FMS Magazine*) 1984, pp 114-117.

Byrd, T.A., and Turner, D.E. "An Exploratory Examination of the Relationship Between Flexible IT Infrastructure and Competitive Advantage," *Information & Management* (39) 2001, pp 41-52.

Carlsson, B. "Flexibility and the Theory of the Firm," *International Journal of Industrial Organization* (7) 1989, pp 179-203.

Chalfant, J. "Quayle Predicts Voinovich Victory," The Associated Press Political Service. September 22, 1990.

Chen, Q. "Understanding the Organizational Impact of Integrated IT Application Infrastructure through Agility: The Case of Enterprise Resources Planning Infrastructure," in: *Management Information Systems: Doctoral Dissertation*, University of Georgia, Athens, GA, 2004, p. 175.

Conboy, K., and Fitzgerald, B. "Toward a Conceptual Framework of Agile Methods: A Study of Agility in Different Disciplines," WISER, Association for Computing Machinery, Newport Beach, CA, 2004, pp. 37-44.

de Groote, X. "The Flexibility of Production Processes: A General Framework," *Management Science* (40:7) 1994, pp 933-945.

Dove, R. "Knowledge Management, Response Ability, and the Agile Enterprise," *Journal of Knowledge Management* (3:1) 1999, pp 18-35.

Dove, R. *Response Ability* John Wiley and Sons, Inc., New York, NY, 2001.

Duncan, N.B. "Capturing Flexibility of Information Technology Infrastructure: A Study of Resource Characteristics and their Measure," *Journal of Management Information Systems* (12:2) 1995, pp 37-57.

Eppink, D.J. "Planning for Strategic Flexibility," *Long Range Planning* (11) 1978, pp 9-15.

Galbraith, J.R. "Organization Design: An Information Processing Perspective," *Interfaces* (3:4) 1974, p 28.

Golden, W., and Powell, P. "Towards a Definition of Flexibility: In Search of the Holy Grail?," *Omega*) 2000, pp 373-384.

Goodhue, D.L., Chen, Q., Boudreau, M.-C., Davis, A., and Cochran, J.D. "Do Enterprise Systems Stifle Agility," (Working Paper) 2008.

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

Grewal, R., and Tansuhaj, P. "Building Organizational Capabilities for Managing Economic Crisis: The Role of Market Orientation and Strategic Flexibility," *Journal of Marketing* (65) 2001, pp 67-80.

Henderson, J.C., and Venkatraman, N. "Strategic Alignment: A Model for Organizational Transformation via Information Technology," in: *Information Technology and the Corporation of the 1990s*, T.J. Allen and M.S. Morton (eds.), Oxford University Press, New York, NY, 1994.

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

Hooper, M.J., Steeple, D., and Winters, C.N. "Costing Customer Value: An Approach for the Agile Enterprise," *International Journal of Operations and Production Management* (21:5/6) 2001, pp 630-644.

CIO Insight. "Who's Spending What on Sarbanes-Oxley?,"(February 1, 2004), accessed on April 5, 2007, from www.cioinsight.com/print_article2/0,1217,a=119467,00.asp

Kaarst-Brown, M.L., and Kelly, S. "IT Governance and Sarbanes-Oxley: The Latest Sales Pitch or Real Challenges for the IT Function?," 38th Hawaii International Conference on Systems Sciences, Waikoloa, Hawaii, 2005.

Klein, B.H. *Prices, Wages, and Business Cycles: A Dynamic Theory* Pergamon, New York, NY, 1984.

Langlois, R.N. "Modularity in Technology and Organization," *Journal of Economic Behavior & Organization* (49) 2002, pp 19-37.

Lee, G., and Xia, W. "The Ability of Information Systems Development Project Teams to Respond to Business and Technology Changes: A Study of Flexibility Measures," *European Journal of Information Systems* (14) 2005, pp 75-92.

Miles, M.B., and Huberman, A.M. *Qualitative Data Analysis* Sage Publications, Beverly Hills, CA, 1984.

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

Overby, E., Bharadwaj, A., and Sambamurthy, V. "Enterprise Agility and the Enabling Role of Information Technology," *European Journal of Information Systems* (15) 2006, pp 120-131.

Patten, K., Whitworth, B., Fjermestad, J., and Mahinda, E. "Leading IT Flexibility: Anticipation, Agility, and Adaptability," Americas Conference on Information Systems, Omaha, NE, 2005.

Sambamurthy, V., Bharadwaj, A., and Grover, V. "Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms," *MIS Quarterly* (27:2) 2003, pp 237-263.

Sanchez, R. "Preparing for an Uncertain Future: Managing Organizations for Strategic Flexibility," *International Studies of Management & Organization* (27:2) 1997, pp 71-94.

Shaft, T.M., and Vessey, I. "The Role of Cognitive Fit in the Relationship between Software Comprehension and Modification," *MIS Quarterly* (30:1) 2006, pp 29-55.

Sharifi, H., and Zhang, Z. "Agile Manufactuing in Practice," *International Journal of Operations and Production Management* (21:5/6) 2001, pp 772-794.

Straub, D., Boudreau, M.-C., and Gefen, D. "Validation Guidelines in IS Positivist Research," *Communications of the AIS* (13) 2004, pp 380-427.

Tushman, M.L., and Nadler, D.A. "Information Processing as an Integrating Concept in Organizational Design," *Academy of Management Review* (3:3) 1978, pp 613-624.

van Oosterhout, M., Waarts, E., and van Hillegersberg, J. "Change Factors Requiring Agility and Implications for IT," *European Journal of Information Systems* (15) 2006, pp 132-145.

Violino, B. "New Regs Force Change," in: *Optimize*, 2003.

Wadhwa, S., and Rao, K.S. "Towards a Proactive Flexibility Management View," *Global Journal of Flexible Systems Management* (2:2/3) 2002, pp 1-10.

Wadhwa, S., and Rao, K.S. "Flexibility and Agility for Enterprise Synchronization: Knowledge and Innovation Management Towards Flexagility," *Studies in Informatics and Control* (12:2) 2003, pp 111-128.

APPENDICES

APPENDIX A: CIO/SENIOR IS MANAGER INTERVIEW TEMPLATE

Thanks for taking the time to discuss my research. The information you provide will be confidential and any component used in publication will be stripped of any identifying information, both of you individually and of your organization.

My research is about understanding change in organization information systems due to business needs for change. At the conclusion of the interview, I can discuss more about the research if needed, but for now, I would like to leave it broad so that I don't bias your responses.

1. Do you have a name for the requests for change that your department receives?

2. What types of change requests does your department receive from the business-side? What is the range of changes that your systems undergo? Do you have a categorization scheme for sorting requests? If so, could you elaborate on this scheme, including why it is segmented in a particular way? If not, do you have an informal scheme for categorizing change requests?

3. What IS infrastructure is housed/developed/modified in house? [Based on response] Do you still use mainframe systems? To what extent do you use ERPs? Are you using any SOA?

4. How is your staff structured in terms of handling change? Who manages the process? Who executes the change at the system level? Are there more layers to the change process? How many people are involved in changing the systems?

5. What do you believe are the two biggest barriers to change for your systems when the business requires changes?

6. Can you think of any systems that seem particularly burdensome to change? Why do believe that is?

7. Are any of your systems particularly easy to change? Why?

8. For these examples, does the particular change affect the difficulty? Or are all changes difficult or easy for those systems?

9. In what ways does complexity affect the change process? From what areas does it enter the discussion? The change itself? The system? The staff's ability to interpret? Can these be distinguished, or are they lumped together?

10. For “simple” [or if he has a better term] changes, how well does your IS infrastructure handle changes? Could you cite an example of this type of change and the time/cost to make the change?

11. How about for more complicated changes? Example? Time/Costs?

12. Are there any hidden costs in these changes?

13. Do you have a formal process for handling change requests from the business? What kind of documentation accompanies those requests, both coming in to the department and internal to the department? How are changes tracked, justified, planned, and executed from the documentation perspective?

14. What seems to be the biggest hurdle that your staff must overcome when making changes? Is it complexity, less experience with particular systems or components, being stretched too thin, etc.?

15. Does your department use any metrics to measure change effectiveness/efficiency/costs?

16. At this point, how static do you feel your IS are? Is the system just being tweaked? Undergoing significant changes?

17. What fraction of the change requests do believe are directly related to organizational agility goals? What category do these requests fall into? [If not previously discussed] How well does your infrastructure handle these changes?

18. What areas of the organization are requesting the most changes? Does it differ for the different types?

19. Are any requests ever denied? If yes, why?

APPENDIX B: PROJECT MANAGER/BUSINESS ANALYST/PROGRAMMER INTERVIEW TEMPLATE

Thanks for taking the time to discuss my research. The information you provide will be confidential and any component used in publication will be stripped of any identifying information, both of you individually and of your organization.

My research is about understanding change in organization information systems due to business needs for change. At the conclusion of the interview, I can discuss more about the research if needed, but for now, I would like to leave it broad so that I don't bias your responses.

1. Can you describe to me a system that you work with that has undergone a variety of different changes while you have been here?
   - What do you use the system for?
   - Is it developed in house?
   - How old is the system?
   - How frequently is the system changed?
   - If you have had the opportunity to work on other systems, how would you rank this system compared to others in terms of changes requested?
   - Do you have access to the code?
   - Is the system isolated or integrated with other systems?
   - How modular is the design (at what level)?
   - How complex is the design?

2. Can you describe the different changes that the system has undergone?

3. Taking One example,
   - Do you know the source of the change?
   - What was involved in making the changes?
     - Any coding?
     - Reconfiguration?
   - If the system was well integrated, how extensive were the impacts to other systems?
   - How complex was the change?
   - Did you find complexity at higher levels, or down at the code level?
   - How time consuming/expensive?

4. Taking another example, could you describe it in more detail?
   - How about the source of this change?
   - What was involved in making the changes?
   - Did this change have impacts on other systems?
   - How complex was the change?
   - And the source of the complexity?
   - Time and Cost?
   - How would you characterize this change as different from the previous one?

- Complexity?
- Difficulty?
- Scope?
- Importance?
- Time pressure?

5. Is there a third example that differs from the previous two changes?
   - How was this different than the others? (continue with the list…)

6. Are there any changes that required rather simple adjustments to the system? Are there particular changes that the system is built to easily accommodate? Do you know if those changes have ever been made?

7. Do you have a categorization scheme for types of changes you encounter? If not, in what ways do you perceive changes differ?

8. What are the two biggest barriers to making changes? Why do you believe those are the most significant?

9. (If they don't mention it) Do you feel the existing system is a burden to change?

10. Have you had to, or considered, significantly expanding the system to meet any change requests?

11. Does your organization/team have a formal process for handling change requests?
    - If so, what is the process?
    - At what level does it occur (high-level, coding)?
    - How effective is the process for getting changes made?
    - If not, what is the default process for making changes?
    - How would you rate your org/team at getting changes done?

Targeted Questions

12. What is the relationship between the change to the IS and the particular system? What in particular makes the system easy or difficult to change?

13. What fraction of changes can be handled by reconfiguration?

14. If reconfiguration is not an option for this system, what are your options for meeting business change needs? Do you have to recode? Do you purchase new systems and integrate with them?

15. To what extent do you believe this system was designed to handle particular business-driven changes? An example of that change? What do you believe are the benefits and drawbacks to designing for business-driven changes?

16. What is the relationship between designing for change and reacting to unforeseen changes?

Personal Profile

17. How would you rate the skill of your co-workers at meeting change demands? How would you rate yourself?

18. How many years experience have you had with the company? In your current role? In IT?

APPENDIX C: PRELIMINARY CODING TEMPLATE

Family	Code	Subcode	Description
Business Changes Impacting the IS			
	Type		Broad categories of business changes that often lead to changes in the IS
		Tax	
		Legislative/Regulatory	
		Merger/Acquisition	
		Add/Remove Process	
		Change Process	
		Reporting	
	Source		
Change to the IS			
	Foreseen/ Unforeseen		Whether the particular change was foreseen prior to development of the software system
	Urgency		How immediate the need for the IS change is
	Diversity		
		Variability	The degree of changes encountered by the IS department
		Variety	The different types of changes encountered by the IS department
		Complexity	The complexity of the changes encountered by the IS department
	Type		The IS department's categorization of change to the IS
	Source		The department/reason for the particular change request
Infrastructure Characteristics Relevant to the Change			
	Technology		
		Level of Modularity	The level of modular design in the particular system (e.g. code, add-on modules, services, etc.)
		Modularity	The degree of independence of various software components
		Integration	The degree of communication between various software components
	Development Process		
		Formal	
		Informal	

	IT Skills		
		Experience	
		Training	
IS Response			
	Type I		The IT department has the option of reconfiguring software to meet change requirements without extensive coding changes.
		Foreseen Changes	
		System Pre-designed for Particular Change	
		Reconfiguration	
	Type IIA		The IT department chooses to extend the functionality of the current system to meet an unanticipated change requirement. This typically requires new coding. Additionally, this may include software not available in the marketplace, or a system they wish to protect.
		Unforeseen Change	
		Strategic / Niche Opportunities	
	Type IIB		
		Expansion of Capability/Functionality	The IT department chooses to purchase software to address a broader need for functionality, which is included in a marketplace package

APPENDIX D: FINAL CODING TEMPLATE

Code	SubCode	Level 3 SubCode	Brief Description
Development Stage			Describes the perspective taken in the development phase with regard to anticipation of IS changes and the change anticipation tactics implemented into applications
Perceived Ability to Anticipate Change			Acknowledgement of the ability to anticipate changes, either business or technical that may impact applications in the future
	Business		Perceived ability to anticipate changes from a business perspective (e.g. change in a business process, business strategy)
	Technical		Perceived ability to anticipate changes from a technical perspective (e.g. changes at the system level)
Perceived Benefit in Implementing Change Capabilities			Acknowledgement of the benefit of implementing change capabilities even if changes are anticipated. May include the difficulty, or cost, or worthwhileness of implementing
Change Anticipation Strategy			Reflects the mindset of anticipating future changes to applications
	Targeted		Attempting to anticipate particular business or technical changes to the application
		Anticipate Particular Business Changes	Attempting to anticipate particular business changes impacting the application
		Anticipate Particular Technical Changes	Attempting to anticipate particular technical changes impacting the application
	General		Attempting to structure or construct the application with general flexibility for changes to the application (i.e. change is coming, it's just unclear exactly which change)
		Build for Broad Flexibility	Constructing applications in a manner where any potential change to the application is prepared for
		Build with Basic Best Practices	Utilizing current best practices in design, such as common languages, OOP, and standards
	No Anticipation Strategy		
Change Anticipation Tactics			

Code	SubCode	Level 3 SubCode	Brief Description
	Modular		Building chunks of the application around logical entites, such as in OOP or around business processes
		Complexity Hiding	Structuring application in chunks that hides complexity that may impact future changes
	Reconfigurable (includes table-driven structure)		Construction of the application where simple modification of parameters changes functionality without having to change at the code level
	Dormant Functionality		Adding functions to application that are anticipated as potential needs but with the functionality disabled
	User Exits		Adding interface points for the possibility of add-on modules or integration points
	Common languages		Using languages that are commonly used and compatible with others
	Other basic best practices (OOP, technical standards)		Using common techniques and standards that are more easily interpreted and interconnected
Post-implementation Stage			Examples of actual occurrences rather than hypothetical examples. When possible, the IS change is coded as anticipated/unanticipated (with the other codes), application characteristics are coded, and the IS response taken is coded
Business Change Affecting IS			A business change that prompts a change or series of changes to an existing application or applications
Business-driven IS Change			A stated action indicating change to a application that is prompted by a business change or requirement
	Type		IS change type (i.e. database change, automated business process change, etc.)
	Local Change		IS change occurs within a particular application
	Interdependency Change		IS change impacts relationships between applications
	Anticipated Change		
	Unanticipated Change		
Application Characteristics Relevant to			

Code	SubCode	Level 3 SubCode	Brief Description
Change			
	Modular		Building chunks of the application around logical entities, such as in OOP or around business processes
	Complexity		
		Complexity - Hidden	Structuring application in chunks that hides complexity that may impact future changes
		Complexity - Not hidden	Complexity serves to complicate a change
		Complexity - Interdependency	Complexity arising from the interdependency between applications
	Reconfigurable (includes table-driven structure)		Construction of the application where simple modification of parameters changes functionality without having to change at the code level
	Dormant Functionality		Adding functions to application that are anticipated as potential needs but with the functionality disabled
	Hard Coding		Inclusion of variables within the application code that requires code modification to change
	User Exits		Places in software that allow for the opportunity to leave one application and interact with a second application, then return to the original application with perhaps some changed values
	Common languages		Standardizing so that applications are built with software languages that are designed to work together
	Other basic best practices (OOP, technical standards)		Using common techniques and standards that are more easily interpreted and interconnected
IS Response			The action taken to accommodate the need for change to an application given the impact of the existing application's structure, design, etc.
	Reconfigure		Application is changed with minimal effort and does not require significant recoding to meet the particular need for change
	Recode		Application is changed by changing the code to meet the particular need for change
		Recode and Restructure (re-enter anticipation phase)	Reorganization of code to both accommodate current change needs, but also prepare for future changes (incorporates change anticipation strategy - development stage) [Coded as response, not

Code	SubCode	Level 3 SubCode	Brief Description
			as a tactic]
	Expand		A new application is acquired or developed outside of the existing applications to meet the particular need for change
		Add on Module	A plug-in module that accomplishes a narrow purpose
		New Application Integration	A new application is acquired where additional functionality is needed beyond the existing application
Also Noting: Places where there is a mismatch between change request difficulty and change implementation difficulty			
Notes:	>Do not code hypotheticals for post-implementation stage		
	>When possible, code IS change, App Char, and IS Response for post-implementation stage examples		