Journal of the Association for Information Systems

Volume 5 | Issue 7 Article 10

7-21-2004

Managing Technology and Administration Innovations: Four Case Studies on Software Reuse

Karma Sherif
Texas Tech University, USA, karma.sherif@ttu.edu

Nirup M. Menon
The University of Texas at Dallas, USA, menon@utdallas.edu

Journal of the Association for Information Systems

RESEARCH ARTICLE

Managing Technology and Administration Innovations: Four Case Studies on Software Reuse[1 2]

Karma Sherif
ISQS, College of Business
Texas Tech University
karma.sherif@ttu.edu

Nirup M. Menon
MSIS, School of Management
The University of Texas at Dallas
menon@utdallas.edu

Abstract

A software process innovation, such as software reuse, involves both technology and administration innovation. Following literature on organizational change, absorptive capacity, innovation assimilation stages, and software reuse, we develop a process model of the assimilation trajectory of an organization's innovation. The model postulates that actors at different organizational levels implement strategy, process, and culture changes in order for an organization to advance through the stages of innovation assimilation. The actions at these levels instill routines that establish the absorptive capacity for implementing future innovations. Case-study data collected from four software development sites – two reporting failure in the reuse program, and two reporting success – revealed that programs that implemented change at the strategy, process and culture levels scored higher on all paths in the model than non-successful programs. The right incentives help in the latter stages of innovation assimilation during which culture change by operational staff is important.

Keywords: Process innovation, Technology innovation, Software reuse, IS development, Case research, Qualitative analysis

[1] Robert Zmud was the accepting senior editor for this paper; Arun Rai and Anne Massey were blind reviewers for this paper.
[2] Karma Sherif thanks Texas A&M for funding the research. The authors thank Glenn Browne, Robert W. Zmud, and seminar participants at Texas Tech University for their feedback on the paper. The research assistance of Ms. Yan Liu is appreciated. Both authors contributed equally on this paper.

Introduction

An innovation is the implementation of a new idea to maintain or enhance organizational performance (Damanpour, 1987). A software process innovation “results in a change to the organizational process used to develop software applications” (Fichman and Kemerer, 1997, p. 1348). Software reuse is a software process innovation in that it shifts the focus from developing software artifacts for a single product to developing them for a future family of applications (McCain, 1991; Basset, 1997; STARS, 1996b; Fichman and Kemerer, 2001). Reuse changes the “throughput technology” in software organizations (Bigoness and Perreault, 1981; Ettlie and Reza, 1992; Hatch and Mowery, 1998). In addition to technological changes, reuse also requires a change in the structure and administrative processes of software development organizations (Apte et al., 1990; Ravichandran, 1999).

This research reports on the success of reuse assimilation in the software development process (Frakes and Fox, 1995; Kim and Stohr, 1998; Rine and Sonnemann, 1998). Following research on technology and administration innovation (Damanpour, 1991), innovation assimilation stages (Cooper and Zmud, 1990), organizational change (Orlikowski, 1993; Eisenhardt and Martin, 2000), and absorptive capacity (Cohen and Levinthal,1990; Zahra and George, 2002), we present the theoretical basis underpinning the assimilation of process innovations. We then develop a testable research model as applied to the context of software reuse. The premise of the model is that purposeful actions in organizations are necessary to influence the assimilation stages of a process innovation (Markus and Robey, 1988), and different organizational actors (or levels) play different roles as the innovation advances through the stages. These actions--collectively named based on organizational levels--are: strategy change, process change, and culture change. Strategy change refers to the business-level policy and resource modifications typically made by senior-level managers. Process change deals with the process- and technology-level changes made by middle-level or project-level managers, and culture change refers to the changes in the attitudes and beliefs of operational staff. The ease with which an innovation is assimilated depends on the capability of the organization to “process” innovations, or the organization’s absorptive capacity to acquire, assimilate, transform, and exploit knowledge in different contexts (Zahra and George, 2002). Strategy change, process change, and cultural change contribute to establishing a state of absorptive capacity that encompasses routines and skills, which in turn help the organization assimilate future innovations.

The research design is a case study design. The data collected enables us to test some necessary conditions for the assimilation of reuse. A case study research design also enables discovery of other facilitating factors from a process view. Though researchers have reported a number of difficulties associated with reuse adoption, empirical investigation of these factors has been lacking (Biggerstaff and Perlis, 1989; Hooper and Chester, 1991; Frakes et al., 1991; Card and Comer, 1994; Jones, 1994; Kim and Stohr, 1998; Fichman and Kemerer, 2001). This study fills that gap. Multiple respondents, representing various levels of decision-making in multiple organizations, provided input on organizational changes that affected reuse adoption at their sites. The findings of our research indicate that effective strategy, process, and cultural changes are all necessary to a certain degree, but not sufficient to ensure the success of software reuse programs. The data shows that sites with inadequate process change implementation were still able to assimilate reuse when the entrepreneurial reuse expert provided developers with the incentives via client participation.

The paper is organized as follows. In the next section, we present our conceptual model that links existing theories. From this conceptualization, we derive the assimilation of software reuse and a set of hypotheses that specify relationships between organizational factors and success of reuse implementation. Following that, section 3 describes the case studies and the data analysis. In section 4 we discuss implications of our findings, and section 5 summarizes the contributions of the research.

The Conceptual Model

Assimilation is "the success achieved by firms in utilizing the capabilities of [their technologies and processes] to enhance their business performance" (Armstrong and Sambamurthy, 1999). Assimilation of a process innovation requires that people manage the change that accompanies the innovation. Indeed, the Lewin model for managing change is the basis for the stages of innovation assimilation – initiation, adoption, adaptation, acceptance, routinization, and infusion (Cooper and Zmud, 1990). In the initiation stage, managers scan the internal and external environment for new innovations to add value to the organization. In the adoption stage, managers make changes in organization structure, such as adding new positions so that there is the necessary backing for the innovation, and redefine organizational processes. In the adaptation stage, the organization is ready for the innovation. The acceptance stage is a milestone that indicates success because employees actively use the innovation. Routinization occurs when the innovation becomes institutionalized and processes and incentives are built around it. The final stage of innovation assimilation is infusion, indicating that an organization uses the innovation to its fullest potential, possibly exceeding initial estimates of its potential. Based on the above conceptualization of innovation assimilation, reuse programs are successful when the organization has advanced to the final stage of infusion,fully utilizing the capabilities of reuse technology to enhance software development productivity.

Various organizational actors impact the progress of the innovation through the assimilation stages by instigating the appropriate changes. Senior managers, middle managers, and operational staff are all involved in various capacities in each of the stages of innovation adoption depending on the prevalent mode of decision-making and entrepreneurial actions – top-down, bottom-up, or middle-out – in the organization.[3] For example, in the initiation stage, senior or mid-level managers or even operational staff may scan the environment and make suggestions, but only the senior management can provide strategy direction. During each stage of assimilation, one level or role dominates. For example, senior managers articulate and support strategy changes (Orlikowski, 1993; Leonard-Barton, 1991; Orlikowski and Hofman, 1997): these changes are necessary at the initiation stage of the innovation. As the organization begins the adoption stage for the innovation, and attempts to appropriate the innovation to better suit its own environment, mid-level managers introduce and execute changes in processes. Likewise, the routinization and infusion stages can only be sustained by culture changes at the operational level (Zmud and Apple, 1992). In some organizations, process innovations never get past the initiation or adoption stages, while in others, these innovations reach the infusion stage. Hence, it is important for researchers to pay attention to necessary versus sufficient conditions when examining a phenomenon from the "emergent perspective" that information technology consequences are unpredictable in complex social organizations

[3] Figure 1 should be interpreted so that senior management plays the most important role at initiation stage. If senior management is the only level in the organization that is involved in this stage, it suggests a top-down approach of decision-making and technology adoption in the organization. We are grateful to an anonymous reviewer for suggesting this clarification.

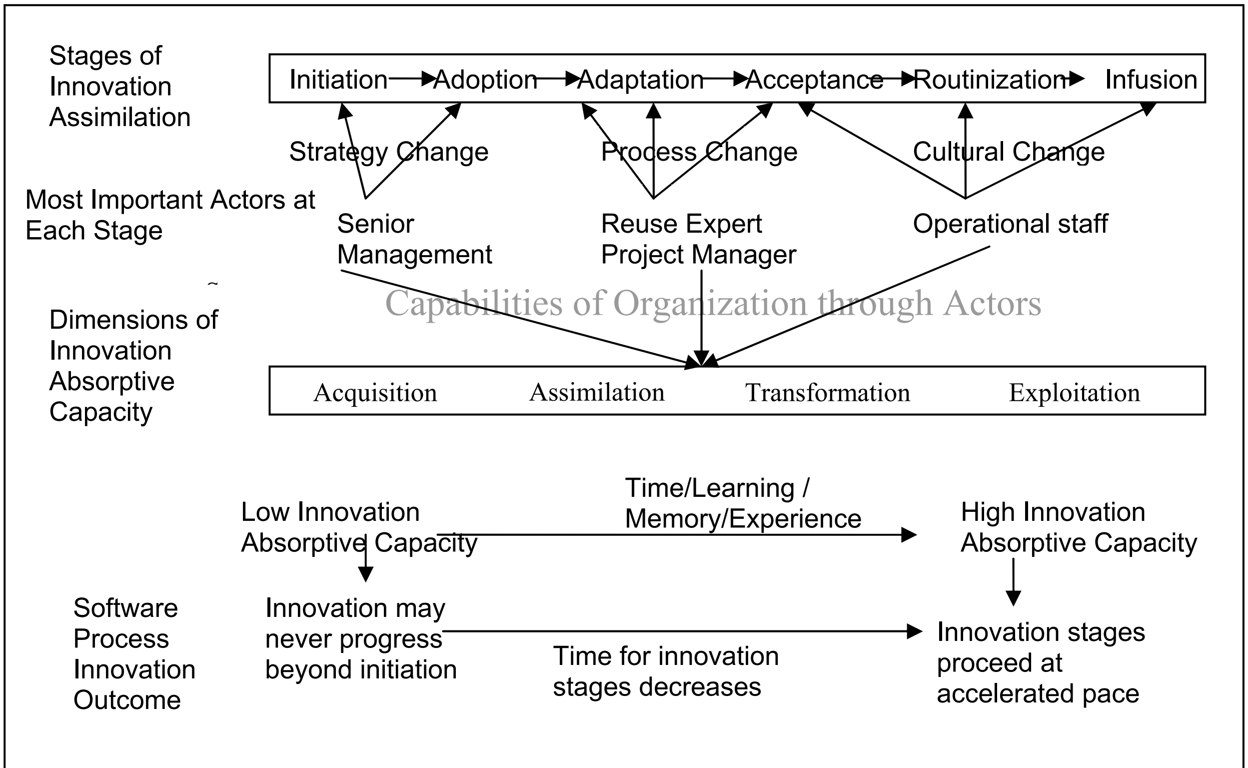

Figure 1. Organizational Roles and Innovation stages trajectory

(Markus and Robey, 1988). For assimilation, some changes are necessary, but not sufficient. In addition, there may be a threshold or degree to which each necessary change must enacted. Still, other factors might be required for the assimilation to be successful, and for routinization and infusion to occur (Zmud and Apple, 1992).

The ability of the organization and its actors to enact the necessary changes is a function of its innovation absorptive capacity (Cohen and Levinthal, 1990; Eisenhardt and Martin, 2000; Zahra and George, 2002). Absorptive capacity is a state or a set of routines that describe an organization's readiness for learning. In our present context, these routines relate to how process innovations are assimilated. The main dimensions of absorptive capacity are acquisition (of facts), assimilation (imbibing knowledge), transformation (using knowledge), and exploitation (generating new knowledge). The corresponding capabilities and skills honed are learning, understanding, conversion, and implementation (Zahra and George, 2002).

Though the earlier conceptualizations of absorptive capacity do not contain a sufficiency condition between the dimensions, an organization must cross a threshold for the assimilation and acquisition dimensions before it can excel in the transformation or exploitation dimensions. This also follows from the fact that absorptive capacity changes with time, and the organization, or more accurately the organizational actors, are learning routines that have succeeded in the past in a similar context. The more experiences organizations have with process innovations (e.g., CASE, object-oriented, reuse), and the better skilled individuals they have at strategic, process and tactical levels, the quicker the organization escalates on the absorptive capacity scale. Developing such a process view of absorptive capacity will be useful in further understanding learning and change processes in organizations.

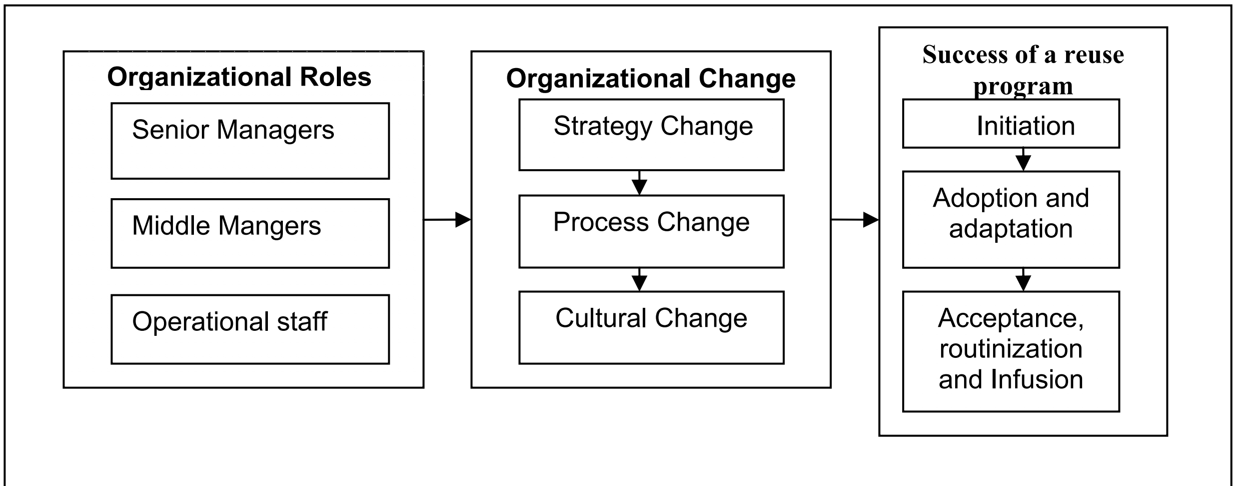

Figure 2. Organizational Roles, Change and Success of Reuse Programs

Innovation assimilation is a good example of how to view the state of absorptive capacity through the process view. Innovation assimilation is the trajectory of a particular innovation, with some overlap between the stages (Cooper and Zmud, 1990). An organization with the necessary level of absorptive capacity will quickly move through the stages of innovation assimilation, while one lacking the necessary level of the absorptive capacity state will take longer, or might ultimately fail to assimilate the innovation (Fichman and Kemerer, 2001). Figure 1 maps the stages of innovation assimilation and the dimensions of innovation absorptive capacity through organizational actors. The success of an innovation depends on how effective organizational actors are in learning to take the correct actions based on established routines. That is, success comes when organizations build the absorptive capacity necessary for them to advance through the assimilation stages of new changes or innovations. This absorptive capacity exists through the collective learning, memory, and experience of the actors at strategy, process, and tactical levels. Each level of management and operation plays a different role in the assimilation of the innovation as well as in developing absorptive capacity. For example, learning in terms of environmental scanning, which is an important aspect of the acquisition dimension, is a prime responsibility of senior managers. Similarly, exploitation-related routines must be established and followed by operational staff, especially in the infusion stages of the innovation.

A Model of Reuse Success

The conceptual model proposed above yields several testable models, and we present one that relates organizational roles, organizational change, and the success of reuse initiatives (Figure 2). Like other software process innovations, software reuse is a major organizational change initiative. Successful implementation of a reuse program requires a change in strategy (e.g., more flexibility of products and less time to market), change in processes (for developing and integrating reusable assets), and change in developers' attitudes (to share and reuse).

Strategy Change

Strategy change encompasses the business-level decisions that govern innovation implementation. Developers face high learning costs and knowledge barriers in the early stages of reuse adoption. But strategy change – administrative guidelines, resource allocation, education and learning programs, and reuse-specific organizational positions – compensates for these learning costs by providing appropriate incentives for project managers and developers. Hence, once strategy change occurs, the organization is able to move successfully ahead on the innovation stages. Thus the changes in administrative guidelines (Harter et al., 2000) and corporate strategies made by senior management (Orlikowski, 1993; Boynton et al., 1994; Leonard-Barton and Deschamps, 1988; Kwon and Zmud, 1987) are success factors in the assimilation of innovations.

Appropriate allocation of resources is also critical to the success of process innovations (Kwon and Zmud, 1987; Boynton et al., 1994; Ravichandran and Rai, 2000). In order for a reuse program to develop and manage reusable assets as well as market and maintain support for the program, significant resources are required. (Poulin, 1997). Additional resources provide incentives to reusable asset and application developers to *build customized solutions for and with* reuse (Ravichandran, 1999; Fichman and Kemerer,2001). Without resources, it is unlikely that a reuse program would advance beyond the initiation stage.

Senior management determines the scope of change, and also delimits the processes and individuals affected by the change. Sometimes organizational change is evolutionary (or iterative), and at other times, it is radical. In cases where the change is complex and results are unpredictable, an evolutionary (iterative) mode is less risky (Benjamin and Levinson, 1993; Cho and Kim, 2002; Harkness et al., 1996; Orlikowski and Hofman, 1997). Because reuse is a "paradigm shift" (Frakes, 1994) involving steep learning costs, the evolutionary approach to implementation yields better results than a radical approach (Jacobson et al., 1997; STARS, 1996a). In the evolutionary approach to reuse, early deliverables of the software development lifecycle are reused, in addition to software code (Frakes, 1994). While reusing analysis models and design patterns has less impact on the cost and time of development, it reduces the risk of wasting reusable assets due to technological change. An evolutionary approach lowers the pressure of adopting new development processes for project managers and developers.

Because the knowledge being reused is so complex, all stakeholders in reuse must have a planned education and training program (Hoopers and Chester, 1991; STARS, 1996a; Jacobson et al., 1997). Necessary for achieving the goals of a change initiative (Agarwal and Prasad, 2000), education and training are best initiated at the very beginning of the initiative so as to help lower the associated knowledge barrier. Well-informed individuals are more likely to understand the need for the change, and hence more likely to support it.

A change agent is critical for managing and sustaining change in organizations (Markus and Benjamin 1996, Scott and Vessey 2002). Management can signal the importance of reuse by creating a formal reuse expert position. The reuse expert becomes the change agent who fosters reuse across different application development projects and builds momentum to sustain the reuse program. A reuse expert also influences software developers' attitudes toward reuse so that they will voluntarily adopt reuse processes (Jacobson et al., 1997). Through mentoring, coordination, and negotiation, the reuse expert improves the organization's capacity to assimilate reuse at both the project and individual level.

An organization that has effectively put the necessary changes noted above in place, will effectively advance through the innovation assimilation stages. By establishing the right mechanisms and routines for these changes in the early learning mode, the organization also builds innovation absorptive capacity. Thus, the necessary condition for future process and culture change is an the adequate degree of strategy change. Hence, the following hypothesis –

Hypothesis 1: Strategy change through administrative guidelines, allocation of resources, scope of change, education and training programs, and change agents, is necessary for a reuse program to advance forward through the stages of assimilation.

Process change

The onus of making the organization able to understand, convert, and internalize innovation lies with the project managers and reuse experts. They enact process changes such as the methods for implementing new processes, the tools made available to support the process (Boynton et al., 1994), the degree of coordination between the various stakeholders involved (Adler, 1990; Scott and Vessey, 2002; Chatterjee et al., 2002), and the measurement and evaluation of progress (Kwon and Zmud, 1987; Dean and Bowen, 1994; Garvin, 1998; Ravichandran and Rai, 2000). Using resources provided by senior management, the project manager allots developers time to build and use reusable assets.

Process change encompasses methods- and technology-related change. A project manager who adopts a systematic and disciplined technique for developing a reusable component will see better software products that have few defects and exhibit high adaptability (Brownsword and Clements, 1996). Thus, process change requires a well-defined reuse methodology in the software development cycle (Jacobson et al., 1997). Automated tools such as CASE tools (Basili et al., 1996) and software repositories (Mili et al., 1995) are reported to help in reusable asset creation, search, and consumption. However, before technology can be deployed to help, project managers must structure the software development process to incorporate reuse. They should also strongly emphasize measurement in reuse such as the assessment of costs, time of development, and product quality (Fenton, 1994; Withey, 1996; Rothenberger and Dooley, 1999).

Finally, reuse activity requires coordination between the reuse group and the software development group. This is true even if the creation of assets and their use in other products is separated by a period of time. In an ideal situation, the reuse team interacts actively with software developers to build a repository of reusable assets. Synergy from team-based structures is also a necessary ingredient in change initiatives (Kwon and Zmud, 1987; Guha et al., 1997). Good communication and coordination among the different stakeholders positively influences the implementation of an innovation. Based on the above discussion, we hypothesize that:

Hypothesis 2: Process change, through structured implementation methods for the development and integration of assets, tools for developing and storing assets, process measurement, and coordination between reuse stakeholders, is necessary for a reuse program to advance forward through the stages of assimilation.

Culture change

Culture refers to the organization's operational work systems (Zmud and Apple, 1992), or the collective practices, values, and beliefs held in the organization (Detert et al., 2000). A significant anteceding factor for change implementation is that the environment in the organization must be receptive to change (Tushman and Nadler, 1986). Whether innovation is optional or mandated, users' response is critical so long as they have the power to delay, obstruct, or underutilize a new technology (Kimberly and Evanisko, 1981; Leonard-Barton and Kraus, 1985). A change in culture will require a shift in "the attitudinal and behavioral stance taken within an organization by targeted users of an innovation" (Leonard-Barton and Deschamps, 1988, p. 604).

"The attitudinal and behavioral stance" of operational staff becomes important in the last three stages of assimilation innovation: acceptance, routinization, and infusion. In the acceptance stage, those who will ultimately use the innovation must be induced to do so. This is the stage in which the operational staff's biases and political resistance (Markus 1983) are addressed. Later, during routinization and infusion, staff use the innovation without much effort, and it is optimized along organizational workflows and used at the full potential. A major cultural challenge for the acceptance, routinization, and infusion of reuse is that software developers may be "possessive" of their code. Some developers act as though they own the code they have created and try to keep others from altering it (Frakes, 1994; Jacobson et al., 1997). Software developers may also oppose using code not developed in-house, adopting the "Not Invented Here" (NIH) attitude (Hooper and Chester,1991; Kim and Stohr, 1998) and constrain reuse to software code that is opportunistically hunted and gathered (Banker and Kauffman, 1991). In summary, operational staff must exploit the innovation in the infusion stages until they are able to use it "even in their sleep." By putting the mechanisms and routines into their work processes, they build the innovation absorptive capacity of the company. Hence,

> **Hypothesis 3**: Culture change, exhibited through software developers' favorable attitude toward reuse and active sharing of their knowledge, is necessary for a reuse program to advance forward through the stages of assimilation.

Case Research Design and Execution

We adopted a multi-site case study approach for replication logic (Yin, 1994) and to test the validity of the proposed hypotheses (Eisenhardt, 1989; Benbasat et al., 1987). The unit of analysis was the reuse organization unit at four different organizations (without focusing on any one software development project at any site). We identified the stakeholders as: *senior managers* who oversaw the entire software development process; *project managers* who were in charge of one or more software development projects; *the reuse expert or reuse champion*, an individual or group who fostered the idea of reuse and sought to institutionalize reuse across the organization; *asset creators*, software developers who were specifically responsible for the development of reusable components; and finally, *asset utilizers,* software developers who primarily reused assets during development. We interviewed at least one respondent from each category at each site to determine their perceptions along the three constructs of the model. Each was or had been involved in at least one project that handled both the creation and use of reusable assets. We encouraged them to speak about their cumulative experience on reuse at their organization, and about the facilitating and inhibiting factors of the reuse program, guided by the questions on Table 1. All respondents provided input to questions on strategy, process, and culture change.

We recorded and transcribed the interviews and applied the semiotic mode of analysis following the Krippendorf (1980) approach of "content analysis," assigning words from the interviews to an indicator from the proposed model. Data analysis focused on analyzing the interviewees' comments (subjective interpretations) along the indicators (researchers' interpretations) that define each construct (see hypotheses 1-3). We assessed the level of importance of a dimension based on the frequency with which it appeared in the text of the transcribed interviews.

Site Selection

We identified and contacted thirty-five different organizations and conducted a preliminary interview with at least one individual from each of these organizations. The interview sought to determine the degree of formality and structure of the reuse program. We then applied three criteria in the selection of sites. First, the organization should have a formal reuse program with specific goals and objectives. Next, there had to be a clear distinction between the development of software applications and the systematic creation of reusable assets. Finally, there had to be a formally defined role for a reuse expert. The reason for this is that, rather than compare highly unstructured reuse initiatives with highly structured initiatives, we intended to compare failed implementations with reasonable organizational interventions for reuse, with successful implementations with similar interventions. We identified four sites (companies or subsidiaries)– two that reported successes, and two that experienced failures of the reuse program, according to the initial interviewee. We refer to the four sites below as site 1, site 2, site 3, and site 4 to protect their identity.

Site 1 is an information technology services company that uses client/server technologies and specializes in providing solutions to external clients in industries such as energy, communications, financial services, and the government. Its scope of reuse encompassed several domains. We interviewed five reuse stakeholders: the project manager, the reuse expert,[4] two developers as asset creators, and one developer as an asset utilizer. Developers at site 1 believed in reuse as the means for developing quality systems, and senior management financed the development of a reuse infrastructure (methods, tools, and training). Individual projects financed the development of the reusable assets, and the organization established a well-defined set of processes to govern the development and use of reusable assets. The methodology for system development rigorously supported reuse through all phases and was coupled with a framework of reusable components that standardized operations across projects to speed up development. The group was successful in compromising between building for reuse and delivering to external clients on time through iterative development of reusable assets. Everyone interviewed agreed that the reuse initiative was successful.

Site 2 is a Customer Billing Services (CBS) unit at a large communications and information services company with annual revenues of several billion dollars. CBS provided billing services and customer care applications for residential customers. Hence, the domain for reuse was primarily in billing and sales applications. We interviewed eight reuse stakeholders within CBS: two reuse experts, the manager of the CBS unit, a project manager, two asset creators, and two asset utilizers. The initial interview and subsequent interviews confirmed that the reuse initiative at site 2 had failed. Management took the initiative of setting formal guidelines for implementing reuse, but admitted to a lack of discipline in reuse. They had set aside a limited budget for the development of reusable assets; however the majority of the funding came from the individual projects. Several reuse stakeholders conveyed an unfavorable attitude toward software reuse. The reuse expert found it "…very difficult to get people to adapt to it." Respondents did not think that reuse was "…at the top of…a division's or directorate's goals list right now."

Site 3 is an oil and gas company with a worldwide presence. The IT organization at site 3 is a subsidiary based in the U.S. with commercial revenues of several million dollars. The IT organization started a reuse initiative in 1996 that covered projects in procurement, production, and sales within the enterprise. We interviewed twelve stakeholders at site 3: three reuse experts (the reuse expert at production and management, and the previous and the current

[4] The reuse expert was also the director of the unit.

reuse experts in oil exploration and production), three asset creators, four asset utilizers, one project manager, and a senior executive knowledgeable about reuse. The reuse stakeholders at site 3 believed that reuse, as a technology, is capable of achieving its theoretical benefits. But managers and developers had no enthusiasm for reuse. The unit had tried to systematize reuse twice, but failed, spending $2.5 million on the last initiative alone. All developers interviewed believed that reuse is a "complex effort" that is difficult to implement, and they attributed the failure to technology and tools. In addition, they believed that it was hard to create reusable assets and difficult to integrate them within applications because of interdependencies. Their notion of reuse was more opportunistic than systematic in nature, and was not tied to the software development process.

Table 1. List of Questions for the Interview

Theoretical Construct	***Question***	***Indicators***
Strategic Change	1. Are there any policies or strategies that govern reuse? 2. Are you aware of any policies or strategies that unintentionally constrain reuse?	***Administrative Guidelines***
	3. What resources are allocated to the reuse program? 4. Are you rewarded for reuse-related activities? 5. Are there education and training programs specifically targeted to reuse?	***Resources***
	6. How did you make the transition from custom development to reuse-oriented development? 7. How are the assets created and utilized?	***Scope of Change***
	8. Is there a reuse champion who fosters the reuse program?	***Change Agent***
Process Change	9. Do you follow a structured methodology for creating reusable assets? 10. Do you follow a structured methodology for integrating reusable assets?	***Structured Implementation Methods***
	11. Do you have tools to create and integrate assets? 12. Do you have repositories to store the reusable assets?	***Tools***
	13. Do you collect reuse metrics?	***Process Measurement***
	14. How do you communicate with other reuse stakeholders?	***Team-Based Structures***
Cultural Change	15. Do you believe software reuse to be beneficial to the organization? 16. Do you believe it is easy to create reusable assets? To locate these assets? To understand these assets? To integrate these assets in other applications?	***Individual attitudes toward reuse***
	17. Are people in the organization willing to share their experience freely with others?	***Knowledge Sharing***

Site 4 is a leading software consulting organization that delivers systems to external clients in several industries. The site's Energy Solution Group provides mainly one product, a gas accounting system, and its reuse domain focused on financial systems and external and internal accounting. We interviewed five stakeholders at this site: a reuse expert who also served as the director of the group and the architect of the reuse framework, two asset creators, and two asset utilizers. For the Energy Solution Group at site 4, reuse is an integral part of the software development life cycle. Their goal is to abstract the whole gas accounting system to the extent that developing new systems would require just a change in the settings of a database, allowing one client's version to be just a reproduction of the base features they have modeled. Like site 1, IT management provided the funds for the development of reuse infrastructure, while individual projects financed the development of the assets. Management encouraged developers to create new applications using prefabricated building blocks, offering no choice for the asset utilizers to reuse or build from scratch. They also instituted a review process to catch situations when reuse could have been employed. Site 4 focused not only on reuse of assets, but also on reuse of knowledge and project experience. A summary of the characteristics of the 4 sites is presented in Table 2.

Table 2. Summary of Reuse Program Characteristics

Descriptive data	Site 1	Site 2	Site 3	Site 4
Industry	IT Consulting	Telecommunication	Oil and Gas	IT Consulting
Organizational Structure and reuse team	Reuse team is part of software development.	Reuse team is part of software development.	Reuse team is separate but in IT unit.	Reuse team is a separate entity from IT unit.
Financing of the reuse program	Management supported the development of an infrastructure (education and training, structured methods, and tools). Individual projects supported the development of the reusable assets (staffing).	Management provided limited support for the development of an infrastructure (structured methods, and tools). Individual projects supported the development of the reusable assets (staffing).	Management provided full support for the development of an infrastructure at the early stages of the reuse project (structured methods, and tools, staffing). Management pulled out resources when it couldn't reap the benefits of its $ 3 million investment.	Management supported the development of an infrastructure (education and training, structured methods, and tools). Individual projects supported the development of the reusable assets (staffing).
Relationship of asset creators to Application Development	Asset creators are appointed to reuse-oriented development projects	Asset creators work part-time on developing assets	At the initial stage of the program, asset creators worked full time developing assets then converted to the part-time model.	Asset creators are appointed to reuse-oriented development projects.

Data Collection

We conducted, taped and transcribed a total of 30 interviews within the four organizations (Table 3). Prior to the interviews, we collected archival data in the form of articles, promotional material, and Internet web pages about the companies and their reuse programs. This supplemental material provided face validity for the questions posed during the interviews and enabled the researchers to comprehend the responses in their appropriate contexts.

Table 3. Number and Type of Respondents (Interviews) at Each Site

	Site 1	Site 2	Site 3	Site 4
Reuse Expert	1	2	3	1
Asset Creator	2	2	3	2
Asset Utilizer	1	2	4	2
Project Manager	1	1	1	--
Senior Manager	*	1	1	*
Site Total	5	8	12	5

*: The reuse expert also served as the director of the unit where we interviewed.

Data Analysis

We read each transcribed interview carefully, and coded data (phrases in respondents' statements) for the indicators (Carlson and Davis, 1998) using QSR NUD*IST software to dissect the interviews. Respondents commented on the presence and absence of changes in both the affirmative (facilitator indicator) and the negative (inhibitor indicator) senses. We coded the presence of a facilitator and the absence of an inhibitor in comments as positive counts for the indicator as defined by the researchers. Alternatively, we coded the presence of an inhibitor and absence of a facilitator as negative counts (see Appendix B for examples of positive and negative quotes). Thus, we derived the positive and negative counts for all indicators for each site (Table 4). Two researchers, of which one was blind to the hypotheses of the study, performed the coding independently based on a set of cues (see Appendix C). The coding resulted in an inter-rater reliability of 82%.

We analyzed the interview data in two ways. First, we summed the positive and negative counts for the various indicators for each site for the respective indicators and constructs per the model in Figure 2. We computed the ratios of positive to negative counts for each indicator for each site (Table 4). This enables visual comparison of the relative numbers of positive and negative comments. We also conducted a non-parametric test using contingency tables to assess the predictive power of each indicator toward the outcome of the reuse program.

A problem with such a quantitative content analysis is the assumption that all references to the coded indicators are of equal importance (Weber, 1990). Such analysis also ignores mediating and moderating effects that validate a process view of the phenomenon. Hence, we also conducted a qualitative analysis which was qualitative was based on the researchers' inferences regarding the causal effects alluded to in the interviews (Miles and Huberman, 1994). Qualitative analysis provides a richer description of the causation in the organizational change process, and also helps us unearth other conditions that facilitated the previously determined necessary conditions for success of the assimilation of an innovation.

Research Findings

Visual inspection of Table 4 enables hypotheses validation. The table contains positive and negative counts, a ratio of positive to negative counts for each indicator, and an aggregate factor. The successful sites (sites 1 and 4) showed higher positive to negative ratios on all three constructs compared to the failed sites. Compared to other constructs within the same sites, the positive to negative ratio for the cultural change construct at the failed sites 2 and 3 were the highest, showing that cultural change is necessary but cannot substitute for low levels of strategic or process change. This result differs from the literature on software reuse that concludes that cultural change is the most significant inhibiting factor to reuse. We discuss more detailed findings based on Tables 4 and 5 below.

Table 5 provides the contingency tables non-parametric test for the predictive power of each indicator and construct toward the outcome of the reuse program.[5] This provides statistical robustness rather than judging the truth of the hypotheses by visual inspection. We constructed a contingency table for each indicator and construct and aggregated values for the successful pair of sites and then for the failed pair of sites. Each contingency table consists of two rows (successful and failed) and two columns (positive and negative count) – thus, four cells – for each factor (Conover, 1999). Assuming that the interviewee comments were drawn randomly from a normal distribution, we tested the null hypothesis that the probability of positive counts of an indicator (or higher level concepts) associated with a successful outcome is equal to the probability of negative counts of the indicator associated with a failed outcome. The alternate hypothesis is that the former probability is greater than the latter. The rejection of the null hypothesis would enable us to determine the effectiveness of each factor in predicting the outcome of the reuse (change) initiative. A brief summary of the contingency tables is presented in Appendix A. The Z-statistic larger than the χ^2 value for 1 degree of freedom indicates the significance of the indicator or construct in predicting the outcome of the initiative (Conover, 1999). The data supports all three hypotheses. The high goodness of fit of the overall model is evident from the Z-statistic of 78.1 (Table 5, row for "overall model fit").

Strategy Change

Strategy change is conceptualized as administrative guidelines, allocation of resources, scope of change, education and training, and change agents. Results support that strategy change is necessary for the success of a reuse program (Z-statistic=67.5 at p-value 0.001 in Table 5). Relative to the successful sites, the failed sites (2 and 3) exhibit high negative references on administrative guidelines, and scope of change. These sites evidence a low positive to negative counts ratio on administrative guidelines, allocation of resources, scope of change, and education and training, compared to the successful reuse programs. However, the contingency tables reveal that administrative guidelines and scope of change are highly significant. At sites 1 and 4, the management established structured administrative guidelines that outlined a specific implementation plan to handle the development of reusable assets. At site 1, reuse was perceived as a strategic process that contributes to their equation for long-term value, where value is content divided by time. At sites 2 and 3, there was a lack of formal administrative guidelines to encourage the development and utilization of reusable assets. Stakeholders at both sites believed that management's lack of "real perspective… as to how to leverage reuse across projects," and an overemphasis on time to market led the organization to focus on short-term goals.

Managers at sites 1 and 4 did not establish rewards for reuse because both asset utilizers and asset creators were committed to it. The director of the energy solution group at site 4 explained this view, saying that "monetary incentives for reusing… or any other recognition…would not have any effect on other peoples' jobs." This shows that direct rewards are not necessary for reuse. All four sites reported a shortage of manpower resources for the creation and integration of reusable assets,. indicating that the amount of resources expended on reuse was not a necessary condition. Though management funded the development of the reuse infrastructure at sites 1 and 4, the day-to-day activities of developing new reusable assets or evolving existing

[5] We are grateful to W. Jay Conover for this suggestion.

ones were short of resources. Because resources were lacking, the two failed sites followed a hunter/gatherer approach to reuse focusing on opportunistic rather than systematic development of reusable assets.

The scope of change examines the mode of change from customized development to component-based development. Scope of change at the two successful sites was evolutionary. At both sites, they made a gradual shift to reuse-based development. The assets evolved with every new reuse cycle, thereby spreading the cost of its development across several projects. Site 4 minimized losses from technological changes through the evolutionary development of reusable assets, while site 2 followed the evolutionary approach to building reusable assets, but was opportunistic in its use of the assets. Site 3, on the other hand, followed a revolutionary approach by attempting to implement reuse in a short period of time. This approach faltered because of the lack of reliable support for the integration of assets within projects.

Various respondents highlighted the lack of an adequate reuse education program. At sites 1 and 4, the reuse groups substituted the training program with mentoring between the reuse team and the development team that focused on “learning by doing.” At sites 2 and 3, there were no training programs specifically targeted for reuse. The reuse expert and reusable asset developers understood the technology and the way to implement it, but undertook no effort to educate managers or application developers.

The results support the notion that significance of the reuse expert role is a significant factor in the success of a reuse program. However, the reuse expert role was more significant in qualitative analysis than in the quantitative analysis. When comparing the interviews across the sites, we saw that the four reuse experts enjoyed different levels of authority and used different persuasive powers. At sites 1 and 4, the reuse experts had more authoritative power than their counterparts at sites 2 and 3, serving the dual role of reuse expert and director of the unit we studied within the larger organization. However, the two successful sites used authoritative power differently. At site 4, the reuse expert strongly commended reuse and the integration of assets over customized development, while at site 1 the expert did not exert pressure on the developers to reuse, but pressured the reuse group to prove it was generating value to the application groups.

The reuse experts in sites 1 and 4 both influenced the disposition of external clients toward reuse and convinced them of its strategic importance, working out deals with external clients that helped finance part of the evolution of the reuse library. These reuse experts took the responsibility of educating their external clients about reuse, giving them the option of going with customized development at cost or adapting their requirements to prefabricated solutions, thus benefiting from a quick delivery and a reduced cost. The reuse expert at site 4, as the director of the group, was able to mandate the use of reusable assets in application development without losing the loyalty of the group. Even though sites 1 and 4 experienced pressures similar to those experienced at the failed sites, such as time constraints, the deals with external clients decreased the importance of time constraints, and increased the importance of reuse.

The reuse experts at sites 2 and 3 had limited authoritative power. They both reported to the IS managers of the groups we studied, and they played advisory roles with no power to mandate reuse. They tried to convince developers of the benefits of reuse, but often collided with the developers’ belief that reuse “is not gonna happen” within the organization on a large scale.

Table 4: Counts and Ratios for Indicators and Constructs from Interviews at Sites

Constructs	**Indicators**	**Sites[6]**												**Mean of Ratios for all sites**	**Total -ive count for all sites**
		Site 1		*+ive to –ive Ratio*	**Site 2**		*+ive to –ive Ratio*	**Site 3**		*+ive to –ive Ratio*	**Site 4**		*+ive to –ive Ratio*		
		+ive	*-ive*		*+ive*	*-ive*		*+ive*	*-ive*		*+ive*	*-ive*			
Strategy Change	Administrative guidelines	87	32	2.7	16	71	0.2	107	181	0.6	94	21	4.5	2.0	305
	Resources	39	16	2.4	8	9	0.9	33	38	0.9	8	5	1.6	1.4	68
	Scope of change	20	7	2.9	12	65	0.2	9	54	0.2	18	5	3.6	1.7	131
	Education and training	27	15	1.8	8	28	0.3	20	9	2.2	7	3	2.3	1.7	55
	Change agents	16	13	1.2	4	13	0.3	10	20	0.5	18	9	2.0	0.4	55
	Subtotal	**189**	**83**	**2.3**	**48**	**186**	**0.3**	**179**	**302**	**0.6**	**145**	**43**	**3.4**	**1.6**	**614**
Process Change	Structured implementation methods	49	44	1.1	5	64	0.1	49	100	0.5	30	28	1.1	0.7	236
	Tools	19	20	1.0	2	28	0.1	11	48	0.2	4	17	0.2	0.4	113
	Process measurement	1	4	0.3	0	4	0.0	4	5	0.8	4	3	1.3	0.6	16
	Team-based structure	12	1	12.0	4	0	0.0	5	9	0.6	17	6	2.8	3.8	16
	Subtotal	**81**	**69**	**1.2**	**11**	**96**	**0.1**	**69**	**162**	**0.4**	**55**	**54**	**1.0**	**0.7**	**381**
Culture Change	Individual attitudes towards reuse	40	15	2.7	19	104	0.2	184	147	1.3	135	43	3.1	1.8	309
	Knowledge sharing	44	3	14.7	20	2	10.0	54	34	1.6	42	10	4.2	7.6	49
	Subtotal	**84**	**18**	**4.7**	**39**	**106**	**0.4**	**238**	**181**	**1.3**	**177**	**53**	**3.3**	**2.4**	**358**
	Total	**354**	**170**	**2.0**	**98**	**388**	**0.3**	**486**	**645**	**0.8**	**377**	**150**	**2.5**	**1.4**	**1353**

[6] The sites where the reuse program was reported as a failure are grayed out (sites 2 and 3).

Table 5. Contingency Tables for Testing Significance of Indicators and Constructs

Constructs	S=sites 1 and 4; F=sites 2 and 3*	Sum of +ive counts	Sum of –ive counts	Z-stat.**	Indicators	S=sites 1 and 4; F=sites 2 and 3*	Sum of +ive counts	Sum of -ive counts	Z-stat.**
Overall Model Fit	S	731	320	78.1					
	F	584	1033						
Strategy Change	S	334	126	67.5	Administrative guidelines	S	181	53	55.9
	F	227	488			F	123	252	
					Resources	S	47	21	12.2
						F	41	47	
					Scope of change	S	38	12	55.1
						F	21	119	
					Education and training	S	34	18	10.2
						F	28	37	
					Change agents	S	34	22	14.1
						F	14	33	
Process Change	S	136	123	33.2	Structured implementation	S	79	72	24.5
	F	80	258			F	54	164	
					Tools	S	23	37	16.9
						F	13	76	
					Process measurement	S	5	7	2.4^
						F	4	9	
					Team-based structure	S	29	7	11.3
						F	9	9	
Culture Change	S	261	71	43.1	Individuals' attitudes	S	175	58	37.1
	F	277	287			F	203	251	
					Knowledge sharing	S	86	13	17.9
						F	74	36	

*This column contains the value S if the row cells contain the sum of counts from successful sites (sites1 and 4). It contains F if the cells in the row contain the sum of counts from failed sites 2 and 3.

** distributed χ^2 with degree of freedom of 1.

^Z-stat less than 3.84 indicates $p<0.05$. Rest of the values $p < 0.001$.

The qualitative analysis of the respondents' comments indicated the importance of strategy change during learning and the initial stages of innovation. Participants at the successful sites strongly believed that the resources allocated to reuse helped them fully assimilate the technology (see the following quotes as an example).

> A lot of that work in fostering reuse, we put as part of our management duties as opposed to having it on individual developers' heads because they have other priorities to focus on. We teach our process, our methodology, and also our architecture, and our development standards, we do point out those aspects of those processes and application architecture that is there to support reuse. That's how we really got in the mindset of doing, building reusable components and migrating into a reusable architecture
>
> Senior manager (reuse expert) at site 1

> The standards we have put try to encourage people to develop in ways that are hopefully reusable. The approach we take as far as the process, and the training

program that people go through that teach them the concept of reuse hopefully encourage reuse. From what I have noticed it takes people several iterations as far as reusing certain things. You have to repeat a few concepts a number of times before it sinks in.

Senior manager at site 4

Table 6. Summary of the Strategy Change

Indicators	Site 1	Site 2	Site 3	Site 4
Administrative guidelines were…	Very Structured	Semi-structured	Unstructured	Very Structured
Resources for reuse were…	Adequate	Not adequate	Not Adequate	Adequate
Scope of change was…	Evolutionary	Opportunistic	Revolutionary	Evolutionary
Education and training	Direct mentoring on reuse technology by the reuse expert	Educational courses on reuse technology were taken on an individual bases	Educational courses on reuse technology were taken on an individual bases	Direct mentoring on reuse technology by the reuse expert
Change agents had…	Authoritative powers. Were able to gain the support of application developers and external clients.	Limited authority. Failed to get the support of management and application developers.	Limited authority. Failed to get the support of management and application developers.	Authoritative powers. Were able to gain the support of application developers and external clients.

At the unsuccessful sites, participants believed that management was responsible for the difficulty they experienced implementing reuse because of inadequate actions. The following excerpts highlight the negative effect of inadequate strategic change on process and cultural change.

They lip service through it. They'll say, "Yes, we realize that it's important," but when it comes right down to dedicating people who's full time job is to run a reuse group, that's when it gets difficult….. That's why reuse is not the most immediate thing that people think of when they are doing development. Usually the most immediate thing is I have a deadline, and I need to get it out.

Reuse expert at site 2

Management is apprehensive about investing heavily in reuse. They're apprehensive at this point because of the past track record. For reuse to be successful, it does have to be driven by a management that says, "A focused effort on code reuse is where we want to be, and we're going to take advantage of that." So it has to be driven by management. It also has to have the cooperation of project managers, developers, testers, everybody in the whole development life cycle.

Reuse expert at site 3

Process Change

In process change, we found that structured development methods for reuse, the use of tools to support reuse, and team-based structures were necessary for success (Table 5). One finding that surprised us was that at successful sites, the negative counts and positive counts for these three indicators are close (ratios are close to 1 in Table 4), which might indicate that process change is not necessary according to the component-wise subjective interpretation of respondents. Examination of qualitative data revealed that these sites attempted to apply sophisticated reuse practices and had higher expectations of the outcomes of the reuse program. At the time of the study, standards for component-based development (CORBA and DCOM) were in their infancy; hence, these sites experienced more difficulties in applying advanced concepts. On the other hand, the availability of experienced staff helped in giving reuse the needed boost of confidence.

Structured implementation methods to support the development and integration of reusable assets were vital for the success of reuse. At the successful sites, the established standards defined a set of procedures for the creation and integration of assets. These procedures tied the creation and integration of assets to application development. Furthermore, application development methods were annotated with links to various design patterns and kernels of accumulated best practices in the reuse library. At sites 2 and 3, there were no structured methods for developing and integrating assets. Each project used a separate set of guidelines.

Table 7. Summary of Process Change

Indicators	Site 1	Site 2	Site 3	Site 4
Structured development methods	Advanced	Developing	Undeveloped	Advanced
Tools	Advanced	Developing	Undeveloped	Advanced
Process Measurement	No reuse metrics were collected	No reuse metrics were collected	No reuse metrics were collected	No reuse metrics were collected
Team-based structures	Frequent formal and informal communication	Infrequent formal communication	Infrequent formal communication	Frequent formal and informal communication

The availability and use of tools (or technology) for reuse were necessary for its adoption. There were some complaints regarding the immaturity of the technology and the lack of sophisticated reuse repositories. The failed sites 2 and 3 lacked a central repository with a good search and cataloguing mechanism. Developers maintained that "it was more trouble to find and use than to build on your own." At site 1, they tried to rectify the problem by developing "some catalogues and pattern repositories to disseminate knowledge about the components" to make project managers aware of what was available. The technical actions focused on building a solid reuse infrastructure that supported all phases of developing and integrating reusable assets. For example, at site 1, developers created a layered architecture that is transitive enough to be applied to different situations. Asset utilizers believed that the architecture helped them become good developers because of the framework it provided, especially when they lacked experience in reuse.

Across the four sites, the reuse groups did not formally measure the costs and benefits of reuse. This finding is consistent with Frakes and Fox's (1995) conclusion that the collection of metrics is not common among reuse programs. Companies involved in the study compiled rough estimates of the assets often reused. But some cited the lack of formal reuse metrics as one reason that kept management from investing heavily in reuse, reasoning that, “if you can’t measure it, you can’t manage it.”

The communication and coordination (team-based) efforts, mainly fostered by the reuse experts at sites 1 and 4, were necessary for the success of the reuse programs. Communication ensured that asset creators’ and utilizers’ objectives were aligned. At site 1, at least one asset creator advised development teams on building with reuse. Reuse experts at sites 1 and 4 set up periodic sessions for asset creators and asset utilizers to share knowledge and obtain feedback on the performance of reusable assets. The organization at both sites put sophisticated knowledge sharing tools in place, on top of which various virtual communities built communication webs. Frequent communication between developers helped in aligning their objectives. Embedding asset creators as active participants in development teams helped the creators and the utilizers bridge the gap between developing for and with reuse. At sites 2 and 3, developers complained about the inadequate communication between asset creators and asset utilizers. Their reuse practices lacked maturity, as indicated by the following quote:

> “Basically our biggest problem is that we don't do enough of the up front work. We’re not doing a lot of the modeling correctly. We don't even think when we build the component and check if it’s going to fit within the architecture going forward. We basically have like patchwork architecture all over the place. Everybody’s basically doing their own little thing.”
>
> Asset creator at Site 2

Influencing incentives through beneficiaries down the value chain

The reuse team supports the software development teams in developing software for external and internal clients of the firm. Our study found that direct rewards for reusing artifacts were not necessary; “indirect” rewards provided the necessary incentive for software developers to adopt reuse. The reuse teams convinced the clients for whom the software was being developed of its potential benefits, especially on future projects, and because software developers depend on these clients for repeat business, they willingly adopted reuse upon the clients’ urging. Similar incentives could work for internal clients since theypay their IT department through transfer pricing, and the option to outsource projects is available to them. In the above model, the client-driven incentive structure “pushes” the software development teams to work to lower the learning costs and knowledge barriers associated with adopting reuse. We surmise that the organization was in a better state of innovation absorptive capacity, and understand the need to structure incentives down the value chain.

Culture Change

Individual attitudes toward reuse have the highest total negative count at 309 (last column in Table 3) among indicators at all sites, and is a significant predictor of success or failure (Z-statistic of 37.1 in Table 4). Though the cultural change factor has the smallest negative count compared to the other two constructs, it is a necessary condition for success of reuse. At the successful sites 1 and 4, respondents credited their organizational cultures for the receptiveness to reuse technology. Though they believed developing for and with reuse is challenging, their accumulated experience in building reusable components for a variety of external clients in different industries (recall reuse programs at sites 1 and 4 operated within IT consulting firms) and their mentoring program significantly lowered the learning cost of assimilating reuse processes and enabled them to achieve “economies of scale in learning” (Attewel, 1992).

At sites 2 and 3, asset creators perceived reuse favorably, but managers and other developers were skeptical of its benefits because of the difficulty of integrating reusable assets. They considered reuse an option rather than a requirement. Developers did not perceive that reuse curtails creativity (Not Invented Here syndrome), similar to the Frakes and Fox (1995) study. On the contrary, many developers believed that reuse enabled them to focus on new functionality and extend the reuse platform instead of duplicating work that they or others had done on other projects.

Stakeholders in all four organizations believed that knowledge sharing was important to ensure the cooperation between asset creators and asset utilizers. Though sites 2 and 3 did not have much success with implementing reuse, it was not attributed to knowledge hoarding. Both failed sites had a sharing culture, but this positive trait atrophied due to inadequate attention to strategy and process change. Based on observations at sites 2 and 3, cultural change is not sufficient for the success of reuse.

Table 8. Summary of the Culture Change

Dimensions and coded indicators	Site 1	Site 2	Site 3	Site 4
Individual attitudes towards reuse	Positive	Skeptical	Partly positive	Positive attitude
Knowledge sharing among members	High	High	Medium	High

Implications

The current study has several theoretical implications for management and research of technological and administrative innovations. Type ET generalizability (generalizing from description to theory) from Lee and Baskerville (2003) allows the use of our context-specific and site-specific results to draw the following insights to general technology and administrative innovation assimilation. The ability of an organization to adapt to change requires that senior management provide incentives that counter learning costs for middle management and front-line employees who actually implement the technological or administrative innovation. To do this, senior management must structure the administrative guidelines to direct implementation, resources to finance the development of an infrastructure, education programs to teach and mentor innovative processes, and a change agent who fosters the adoption of the innovation across the organization. These mechanisms, while sending a message to the front-line employees that the innovation is important for the organization, also empower the middle manager to make a shift in focus from the status quo toward the innovation. Middle management carries on the momentum for change through structured implementation efforts, tools, coordination mechanisms, and outcome measurement. The efforts of top management and middle management are particularly needed in cases where there is a wide gap between the status quo and the new innovation (Fichman and Kemerer,1997).

As explained earlier, it is not necessary that change occurs in a top down fashion, with strategic change occurring first followed by process and cultural changes. The data revealed that environment scanning, solution search, and feasibility analysis were conducted either by software directors following a middle-out approach (site 4), or by developers following a bottom-up approach (site 1). Senior management supported the reuse initiative only after a business case, through a number of pilot projects, proved that reuse would generate value for the organization. The success of these projects convinced management and other application developers to commit resources to systematic software reuse with a strategy aligned with the overall business strategy, a process detailing the development and integration of assets, and people to support the technology and integrate it as part of their everyday activities. The following comment from site 4 clarifies this.

> I think probably nothing communicates benefits as well as success, and I can definitely sit down with the developers and go over the pros and cons of component oriented development kind of like I have been doing with you. But it's a lot easier once you have a project that's already using component based development and blowing the hell out of all the other projects because they can deliver solutions for their clients much quicker, much more with greater quality, and with greater certainty. Because the component developers have succeeded, and because the application developers have participated, and because we are realistic, we are much more likely to continue to succeed in the future.
>
> Site 4

An important contribution of our work is that it theoretically ties the stages of innovation assimilation (Cooper and Zmud, 1990) and absorptive capacity (Zahra and George, 2002) using the actions of individuals at different levels in the organization (Figure 3). Senior management, functional managers, and operational staff have different responsibilities to learn and install routines that establish high absorptive capacity. This study addresses one piece of the change and learning behavior of organizations. We propose an integrative picture that relates process views of innovation assimilation and absorptive capacity to the variance view of innovation assimilation of Damanpour and others (Figure 3). The different units of analysis in this area are innovation and organization, and they both should be studied with variance and process approaches. However, the variables and their relationships vary dynamically because of the roles played by individuals in the organization. This study also illustrates the importance of "entrepreneurship" of change agents or innovation champions in enhancing the current state of innovation absorptive capacity. The role of a reuse expert is pivotal, particularly when seeking the buy-in of clients who are the sponsors of the development projects. An innovation champion should seek creative mechanisms to provide incentives to all levels of the organization by influencing different participants (such as customers) in the organizational value chain.

In the above model, the client-driven incentive structure "pushes" the software development teams to work to lower their learning costs and knowledge barriers associated with adopting reuse. This change in channeling incentives is applicable to any technology/administrative innovation that supports the primary value chain of a business. In many businesses, IT plays a supporting role to the main value producing activities. As an extreme example, patients of a hospital could be convinced to use certain hospital IT technologies (example, bed-side physician record keeping), thereby providing physicians the incentive to work against the learning curve of using them in their daily activities. This empirical finding is an important contribution of the study as well.

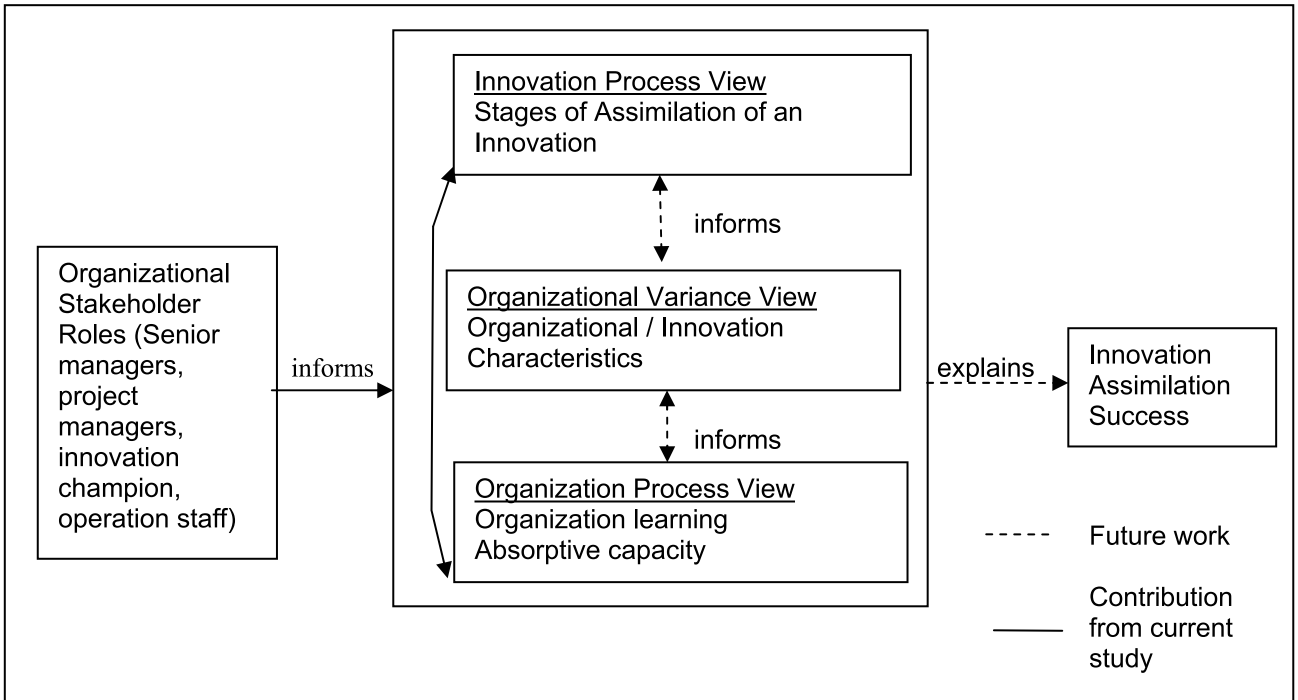

Figure 3. Contribution to Theory

Limitations and Directions for Future Research

Though the qualitative analysis applied in this study has provided in-depth and rich data about reuse practices in four different organizations, future research should further test the conceptual model in Figure 1. An appropriate method would be a longitudinal process-based study that would permit researchers to make conclusions about the direction of causal relationships between strategy change, process change, and culture change, absorptive capacity, and innovation assimilation. We did not operationalize and test the state of innovation absorptive capacity at the sites.

Two of the limitations regarding the framework are: the limited definition of a top management role in reuse adoption and the restricted view of the outcome of a reuse program. We only studied management intervention with regard to administrative guidelines, resource allocation, and educational programs. A more comprehensive view would be to include management's role in resolving political conflicts and managing agency problems among reuse stakeholders. With respect to the outcome of a reuse program, specific measurement of the level and scope of reuse, along with an assessment of the impact of reuse adoption on application development, will be of value to reuse studies.

Conclusions

This paper presents and validates a process model of organizational changes that are necessary for the success and failure of process innovations, using software reuse as an example. We conducted an in-depth case study of four sites, two where reuse was successfully implemented, and two where reuse failed. All sites are leaders in software development. It was found that cultural changes were necessary, but were not sufficient to ensure the success of assimilation of innovation. Strategy change and process change are also necessary for implementing the innovation, but some components as interpreted by the researchers, such as direct rewards and resources, were not necessary. Innovation champions or experts such as reuse experts play a major role in engendering the confidence of senior management, project leaders, operational staff and clients in reuse, by setting up incentives indirectly through the value chain.

References

Agarwal, R. and Prasad, J. (2000) "A field study of the adoption of software process innovations by information systems professionals," *IEEE Transactions on Engineering Management* (47:3), pp. 295-323.

Adler, P. (1990) “Shared Learning,” *Management Science* (36:8), pp. 938-957.

Apte, U., Sankar, C.S., Thakur, M. and Turner, J.E. (1990) “Reusability-based Strategy for Development of Information Systems: Implementation Experience of a Bank,” *MIS Quarterly* (14:4), pp. 421-433.

Armstrong, C. P., and Sambamurthy, V. (1999) “Information technology assimilation in firms: The influence of senior leadership and IT infrastructures,” *Information Systems Research* (10:4), pp. 304-327.

Attewell, P. (1992) "Technology Diffusion and Organizational Learning: The Case of Business Computing," *Organization Science* (3:1), pp. 1-19.

Banker, R.D. and Kauffman, R.J. (1991) “Reuse and Productivity in Integrated Computer-aided Software Engineering: An Empirical Study,” *MIS Quarterly* (15:3), pp. 375-401.

Basili, V.R., Briand, L.C. and Melo, W.L. (1996) “How reuse influences productivity in object-oriented systems,” *Communications of the ACM* (39:10), pp. 104-116.

Basset, P., *Framing Software Reuse: Lessons from The Real World*, Yourdon Press Computing Series, Upper Saddle River, NJ, 1997.

Benbasat, I., Goldstein, D. K. and Mead, M. (1987) “The Case Research Strategy in Studies of Information Systems,” *MIS Quarterly* (11:3), pp. 369-386.

Benjamin, R.. I. and Levinson, E. A. (1993) "Framework for Managing IT-Enabled Change," *MIT Sloan Management Review* (34:4), pp. 23 -34.

Biggerstaff, T. J. and Perlis, A.J. (1989) *Software Reusability: Applications and Experience,* vol. II, ACM Press, Reading, MA.

Bigoness, W. J. and Perreault, Jr. W. D. (1981) "A conceptual paradigm and approach for the study of innovators," *Academy of Management Journal* (24:1), pp. 68-83.

Boynton, A.C. , Zmud, R.W. and Jacobs, G.C. (1994) “The Influence of IT Management Practice on IT Use in Large Organizations,” *MIS Quarterly* (18:3), pp. 299-315.

Brownsword, L. and Clements, P. (1996) *A Case Study in Successful Product Line Development* (CMU/SEI-96-TR-016), Software Engineering Institute, Carnegie Mellon University, Pittsburg, PA.

Card, D. and Comer, E. (1994) “Why Do So Many Reuse Programs Fail?,” *IEEE Software* (11:5), pp. 114-115.

Carlson, P.J. and Davis, G.B. (1998) “An Investigation of Media Selection Among Directors and Managers: From ‘Self’ to ‘Other’ Orientation,” *MIS Quarterly* (22:4), pp. 335-362.

Chatterjee D., Pacini, C. and Sambamurthy, V. (2002) "The shareholder-wealth and trading-volume effects of information-technology infrastructure investments," *Journal of Management Information Systems* (19:2), pp. 7-42.

Cho, I. and Kim, Y. (2001) "Critical Factors for Assimilation of Object-Oriented Programming Languages," *Journal of Management Information Systems* (18:3), pp. 125-157.

Cohen, W. M. and Levinthal, D. A. (1990) "Absorptive Capacity: A New Perspective On Learning And Innovation," *Administrative Science Quarterly* (35:1), pp. 128-153.

Conover, J., *Non-parametric Statistics*, Wiley and Sons, 1999.

Cooper, R. B. and Zmud, R. W. (1990) "Information Technology Implementation Research: A Technological Diffusion Approach," *Management Science (*36:2), pp. 123-140.

Damanpour, F. (1987) "The Adoption Of Technological, Administrative, And Ancillary Innovations: Impact of Organizational Factors," *Journal of Management* (13:4), pp. 675-688.

Damanpour, F. (1991) "Organizational Innovation: A Meta-Analysis of Effects of Determinants and Moderators," *Academy of Management Journal* **(**34:3), pp. 555-591.

Dean J. W. and Bowen, D. E. (1994) "Management theory and total quality: Improving research and practice through theory development," *The Academy of Management Review* (19:3), pp. 392-419.

Detert, J.R., Schroeder, R.G. and Mauriel, J.J. (2000) "A Framework for Linking Culture and Improvement Initiatives in Organizations," *Academy of Management Review* (25:4), pp. 850-863.

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

Eisenhardt, K. M. and Martin, J. A. (2000) "Dynamic capabilities: What are they?" *Strategic Management Journal,* (21:10/11), pp. 1105-1121.

Ettlie J. E. and Reza, E. M. (1992) "Organizational Integration and Process Innovation," *Academy of Management Journal,* (35:4), pp. 795-828.

Fenton, N. (1994) "Software measurement: A necessary scientific basis," *IEEE Transactions in Software Engineering* (20:3), pp. 199-206.

Fichman R. G. and Kemerer, C. F. (1997) "The assimilation of software process innovations: An organizational learning perspective," *Management Science,* (43:10), pp. 1345-1364.

Fichman, R.G. and Kemerer, C. F. (2001) "Incentive compatibility and systematic software reuse," *Journal of Systems and Software* (57:1), pp. 45-60.

Frakes, W. and Fox, C. (1995) "Sixteen Questions about Software Reuse," *Communications of the ACM* (38:6), pp. 75-115.

Frakes, W. (1994) "Systematic Software Reuse: A Paradigm Shift", *Proc. of the Third International Conference on Software Reuse*, Rio de Janeiro, Brazil, pp. 2-3.

Frakes, W., Biggerstaff, W. B., Prieto-Diaz, R., Matsumura, K. and Schaefer, W. (1991) "Software Reuse: Is It Delivering?," *Proc. of the13th International Conference on Software Engineering,* Austin, TX, pp. 52-59.

Garvin, D. (1998) "The processes of organization and management," *MIT Sloan Management Review* (39:4), pp. 33-51.

Guha, S., Grover, V., Kettinger, W. J. and Teng, J. (1997) "Business Process Change and Organizational Performance: Exploring an Antecedent Mode," *Journal of Management Information Systems* (12:1), pp. 119-154.

Harkness, W. L., Kettinger, W. J. and Segars, A. H. (1996) "Sustaining Process Improvement and Innovation in the Information Services Function: Lessons learned at the Bose Corporation," *MIS Quarterly* (20:3), pp. 349-369.

Harter, D. E., Krishnan, M. S. and Slaughter, S. A. (2000) "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," M*anagement Science* (46:4), pp. 451-467.

Hatch, N. W. and Mowery, D. C. (1998) "Process innovation and learning by doing in semiconductor manufacturing," *Management Science* (44:11**)**, pp. 1461-1478.

Hooper, J.W., and Chester, R. O. (1991), *Software Reuse Guidelines and Methods*, Plenum Press, New York.

Jacobson, I., Griss, M. and Jonsson, P. (1997) *Software Reuse : Architecture Process and Organization for Business Success*, Addison Wesley, Reading, MA.

Jones, C. (1994) "Economics of Software Reuse," *Computer* (27:7), pp. 106-107.

Kim, Y., and Stohr, E.A. (1998) "Software Reuse: Survey and Research Directions," *Journal of Management Information Systems* (14:4), pp. 113-147.

Kimberly, J. R. and Evanisko, M. J. (1981) "Organizational innovation: The influence of individual, organizational, and contextual factors on hospital adoption of technological and administrative innovations," *Academy of Management Journal* **(**24:4), pp. 689-714.

Krippendorff, K. (1980) *Content analysis: an introduction to its methodology*, Beverly Hills : Sage Publications, 1980.

Kwon, T.H., and Zmud, R.W. (1987) "Unifying the fragmented models of information systems implementation." In R.J. Boland and R.A. Hirschheim (eds.), *Critical Issues in Information Systems Research*, John Wiley, New York.

Leonard-Barton, D. (1991) "The Role of Process Innovation and Adaptation in Attaining Strategic Technological Capability," *International Journal of Technology Management* (6:3:4), pp. 303-321.

Leonard-Barton, D. and Kraus, W. (1985) "Implementing New Technology**,**" *Harvard Business Review* (63:6), pp. 102-111.

Leonard-Barton, D. and Deschamps, I. (1988) "Managerial Influence in the Implementation of New Technology," *Management Science* (34:10), pp. 1252-1266.

Markus, M. L. (1983) "Power, politics and MIS implementation," *Communications of the ACM* (26:6), pp. 430-444.

Markus, M.L. and Benjamin, R. I. (1996) "Change Agentry-the Next IS Frontier," *MIS Quarterly* (20:4), pp. 385-407.

Markus, M. L. and Robey, D. (1988) "Information Technology and Organizational Change: Casual Structure in Theory and Research," *Management Science* (34:5), pp. 583-599.

McCain, R. (1991) "Reusable Software Component Construction: A Product-Oriented Paradigm," in R. Prieto-Diaz and G. Arango (Eds.), *Domain Analysis and Software Systems Modeling*, IEEE Computer Society Press, Los Alamitos, CA, pp. 132-146.

Miles, M.B. and Huberman, A. M. (1984) *Qualitative Data Analysis: A Sourcebook of New Methods*, Sage Publications, Newbury Park, CA.

Mili, H., Mili, F. and Mili, A. (1995) "Reusing Software: Issues and Research Directions," *IEEE Transactions on Software Engineering* (21:6), pp. 528-562.

Orlikowski, W.J. (1993) "Case Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development," *MIS Quarterly* (20:3), pp. 309-340.

Orlikowski, W.J., and Hofman, D.J. (1997) "An Improvisational Model for Change Management: the Case of Groupware technologies," *Sloan management review* (38:2), pp. 11-22.

Poulin, J. S. (1997) *Measuring Software Reuse: Principles, Practices, and Economic Models,* Addison Wesley, Reading, MA.

Ravichandran, T. (1999) "Software reusability as synchronous innovation: a test of four theoretical models," *European Journal of Information Systems* (8:3), pp. 183-199.

Ravichandran, T. and Rai, A. (2000) "Quality Management in Systems Development: An Organizational System Perspective," *MIS Quarterly* (24:3), pp. 381-415.

Rine, D. and Sonnemann, R. (1998) "Investments in reusable software: A study of software reuse investment success factors," *The Journal of Systems and Software* (41:1), pp. 17-33.

Rothenberger, M.A., and .Dooley, K.J. (1999) "A Performance Measure for Software Reuse Projects," *Decision Sciences* (30:4), pp. 1131-1153.

Scott J. E., and Vessey, I. (2002) "Implementing Enterprise Resource Planning Systems: The Role of Learning from Failure," *Information Systems Frontiers* (2:2), pp. 213-233.

Software Technology for Adaptable Reliable Systems (STARS; 1996a) *Learning and Inquiry Based Reuse Adoption: A Field Guide to Reuse Adoption through Organizational Learning*, STARS Technical Report, STARS-PA33-AGO1/001/02, STARS Technology Center, Arlington, VA.

Software Technology for Adaptable Reliable Systems (STARS; 1996b) *Organizational Domain Modeling (ODM) Guidebook*, Version 2.0, Lockheed Martin Tactical Defense Systems, STARS-VC-A025/001/00, Reston, VA.

Tushman, M., and Nadler, D. (1986) "Organizing for Innovation," *California Management Review* (28:3), pp. 74-92.

Vessey, I., Jarvenpaa, S.L. and Tractinsky, I. (1992) "Evaluation of Vendor Products: CASE Tools as Methodology Companions," *Communications of the ACM*. (35:4), pp. 90-105.

Weber, R. P. (1990) *Basic Content Analysis,* Sage Publications, Newbury Park, CA, 1990.

Withey, J. (1996) *Investment Analysis of Software Assets for Product Lines*, (CMU/ SEI-96-TR-010), Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Yin, R.K. (1994) *Case Study Research: Design and Methods*, Sage Publications, Beverly Hills, CA.

Zahra, S. A. and George, G. (2002) "Absorptive Capacity: A Review, Reconceptualization, and Extension," *Academy Of Management Review* (27:2), pp. 185-203.

Zmud, R.W., and Apple, L.E. (1992) "Measuring Technology Incorporation/Infusion," *Journal of Product Innovation Management* (9:2), pp. 148-155.

Appendix A : Contingency Tables and Their Use in Case Research

In analyzing positive and negative influence of a factor on discrete outcomes (such as successful and failed) outcomes of phenomena, contingency tables, as a non-parametric statistical technique, works well (Conover 1999). Counts of both types of quotes from sites with the different outcomes are tabulated as shown below. S and F are the two outcomes at the two sites, and are represented as the rows of the table. P and N represent positive and negative quotes' counts at the two groups of sites.

	S (e.g., success of phenomenon at sites)	F (e.g., failure of phenomenon at sites)	Total Row count
P (positive quotes on factor)	*a*	*b*	$R_1=a+b$
N (negative quotes on factor)	*c*	*d*	$R_2=c+d$
Total Column Count	$C_1=a+c$	$C_2=b+d$	$M=a+b+c+d$

The statistic in each cell, when divided by total row or column count of quotes, is an estimate of the probability that the factor has in influencing the outcome of the phenomenon.

	S (e.g., success of phenomenon at site 1)	F (e.g., failure of phenomenon at site 2)
P (positive quotes on factor)	p_1	$1-p_1$
N (negative quotes on factor)	p_2	$1-p_2$

Hence, the hypothesis being tested is that the positive or negative perception of the factor is not related to the outcome at a site. That is,

H0: $p_1 = p_2$

H1: $p_1 > p_2$

The Z-statistics calculated from the counts is given by

$$Z = \frac{\sqrt{M}\left(ad - bc\right)}{\sqrt{R_1 R_2 C_1 C_2}} \sim \chi_1^2$$

High values of *Z* (based on the p-value) leads to rejecting the null hypothesis, bearing the conclusion that the factor indeed has a significant influence on the outcome of the phenomenon of interest. The technique can be extended to multiple outcomes and multiple types of quotes (see Conover 1999 for details).

Appendix B: Positive and Negative Excerpts of the Constructs of the model

Constructs	Indicators	Excerpts
Strategy Change	Administrative guidelines	**Positive** For the reuse policies we address things like common approaches for building a reusable component, common architectures, what technologies will make it easy to develop a reusable component, what technologies would make it difficult, how do you train people to develop them and to use them once they've been developed. **Site 1** We were building all these different types of systems, and we had enough experience gained over the years that when a given requirement came in for a system, we could recognize, whether the requirement was specific to that particular system or was that something that was likely to be used on other systems. So we did requirements classification of essentially pinning requirements to the right place in our overall infrastructure. We had three levels of the infrastructure. We had the Foundation, which was used on all systems, and then on top of that we had common systems that were used by many but not all of the individual plant systems and then at the top level, we had the individual specialized plant systems. **Site 3** **Negative** Our primary focus is to build business solutions in time frames or market windows that those business solutions have value to the customers.. Sometimes Reuse is a trade off between delivery time frames and business value and striving for the ultimate reuse. I think that many barriers in my personal page to reuse are tactical focus versus strategic vision. Whenever as an individual or as a team or as a company we tactically focus on something that's going to deliver short term, immediate results like, "We're going to get this project out quick and get the check," we lose our strategic vision to be well respected, high quality providers that have a long future because of the flexibility of the things we build and the quality and the strength of what we deliver**.** **Site 1** The attitude that the organization has, the willingness to sacrifice everything for the time to market, is essentially a constraint. And while it's not a documented policy, for a long time it certainly has been an implicit policy that translates to "I don't really care what you do as long as you get it out when the clients want it or earlier. We have a loose set of guidelines for reuse, but not a focused effort. **Site 3**

Resources	**Positive** Our chances of being a successful reuse program at this moment are pretty good. We have plenty of the intellectual resources needed. There's just a high level of interest throughout the company in maintaining this type of initiative. **Site 1** There was a reuse team responsible for finding the reuse modules, making sure it met standards, went looking for them from off projects, collecting them off, and storing them in repositories. **Site 3** **Negative** It is difficult to convince management to make an investment when looking down 2 years what seems like it might be reusable to you might not actually be if the technology changes so rapidly and the requirements change. **Site 1** People are overworked. They have too much to do, and they can't afford the time to think about reuse while working on projects. The unwillingness of management to pay the extra cost in creating a generic component is a big problem. **Site 3**
Scope of change	**Positive** You do not have quite all the requirements that you need. As the system matures, we have much better understanding of what it is we really wants to build and how best to abstract it. So it is a very iterative process. At this point we realize the number of things we can do better to make it maintainable and abstracted certain business rules abstracting even the technical architecture that we are using to support this thing in such a way that we can more easily adapt to change. **Site 1** We think we got a compromise where the investment for a project to build reusable assets would be minimal because we did not build the complete functionality for the object, we just build the functionality that is required for that specific product or project, but we design it as such that it is extensible, and hopefully that would be the compromise. So the next project that comes along, we could extend that object and put an initial investment in that object and move on from there. So not only is the library evolving, but the objects themselves are evolving from project to project. **Site 3** **Negative** The technology is changing so fast that there is the potential to lose the investment in adopting a technology and make reusable components**. Site 1** It took us three years to populate the repository. The guy that paid for it considered it a tremendous failure because he never saw the savings someone else got the benefits. By the end of the three years period, he had all his software deployed. **Site 3**

	Education and training	**Positive** We develop various case studies that leverage these components as well as various classes in which you learn the component technology, and component remodeling processes. What we're hoping to do is to be able to take new people and churn them through these 4-6 week classes and then have them available as competent developers. **Site 1** We have a learning budget that you can pretty much self-direct. You have lot of freedom to say, "Okay, this is the kind of thing that I'd like to work on this year." **Site 3** **Negative** We do not have a training course solely on reuse. What we do teach though is our process, our architecture, and our development standards, we do point out those aspects of those processes and application architecture that is there to support reuse. **Site 1** It is a pretty lightweight training program, not enough to make reuse happen. **Site 3**
	Change agents	**Positive** Reuse technology alone cannot create a change leader. Reuse must support effective business processes, and people must be prepared to adopt the new technology. …..about half of my day is spent on working with development teams and making sure that their efforts are on time, on track, what issues do they have with the reusable component, also with mentoring, and and helping them out in that way. **Site 1** We set up the whole structure, though, around the fact that we're building this stuff, you are going to be using it. **Site 3** **Negative** One main barrier that I am struggling with is how to make reuse a part of their daily job where they go out, and before they do anything around a particular area they go out and look for things that they can leverage rather than starting from scratch. **Site 1** It didn't work because you didn't have everything, all the requirements together for the right problem we tried to get reuse as a hidden cost and the implicit things are very hard to show. **Site 3**
Process Change	Structured implementation methods	**Positive** Our component frameworks that we've developed facilitate a lot of the main thing that developers have to deal with when developing or integrating components. It include processes and methodologies that support creating reusable components and leveraging reusable components, communications mechanisms for cataloguing components, and how they should be used and what they do. That entire infrastructure must be in place to allow those individuals to be successful. **Site 1** We've recently adopted the RUP development process, and it's perfectly in tune with the object oriented analysis and design which advocate component base development and reuse. **Site 3** **Negative** There need to be a better process to integrate components within projects. **Site 1** Well, the methodology that we use for developing probably hinders it in some way because we don't, we don't really have a, we've tried to institute a more object oriented approach to doing our development methodology, but getting everybody on the same page is a little difficult because of a lack of common terminology across the entire organization. **Site 3**

Tools	**Positive** I don't think that the technology is the biggest issue, but it certainly made things easier for us when everything that we develop follows the same basic architecture, and that architecture made it easy to plug components into applications. **Site 1** I don't think it was a tool problem. We had tools to develop reusable components; we had repositories. Microsoft is making that technology available, making it very easy. They've been pushing in the last few years with their architecture making it very easy to develop reusable components. **Site 3** **Negative** One of our biggest problems is getting developers to know what's available. we're trying to develop some catalogues and pattern repositories to disseminate the knowledge for the components that we're building within the company so that project managers know what's available. **Site 1** If we had some tools that would allow us to keep track of what's reusable or not reusable, it would be great. We need a library, or a dictionary to describe the components so we can easily go and search and see if we can find something close to what we need. **Site 3**
Process measurement	**Positive** I think from a sales perspective it's a little tougher to go in and sell a company based on reuse models. It's probably a lot easier to walk in and demonstrate a piece of functionality to a client and get them to get a very concrete vision of what you can provide them and sell much more quickly based on that. **Site 1** There was an attempt to do metrics, but my personal belief was based on, not necessarily hard data but more observation, looking at the number of components and the number of people that are using them. **Site 3** **Negative** Reuse metrics are not really recorded but I would say that it use to take 2 to 3 days to build a new window. That was about a year ago, now with the architectures and the GUI components it makes it easier to build the window in just about a day. That is kind of hard to pass unnoticed **Site 1** There wasn't any specific measures of the number of reusable components that you built, or reused. **Site 3**

	Team-based structures	**Positive** We value both individual achievement and collective achievement. The ability to work with team members and solve the problem in a team environment is very important to us. **Site 1** They've been pushing very hard for more collective achievement, and part of the evaluation, you know, is on teamwork and team contribution. **Site 3** **Negative** While the project is going on you have billable clients to whom development is being billed. There are times when after a system is being deployed we would not have enough staff to enhance things that we did not have time to implement during development to make it easy for others to reuse. **Site 1** There is far more effort involved in picking a component to be reused on other projects outside the project that originally built it. It's an incredible amount of work, it's an issue of documentation, it's an issue of getting the trust of the people, it's an issue of being successful and supporting. The developers doing the support have to know the code backwards and forwards and they must be supportive. Basically the communication is null unless you interject it. .**Site 3**
Culture Change	Individual attitudes towards reuse	**Positive** Definitely an advantage of doing software reuse is shortening the development life cycle ... It reduces the time to market the product. It stabilizes development to a large extent, and gives them a similar feeling of how the architecture is going to work. So, even if there are multiple applications being developed, if the architecture is the same or similar, then the unit training costs come down. **Site 1** Reuse only gives you an advantage in terms of reducing your software development costs. **Site 3** **Negative** It's actually the quality of the component that's the biggest problem. And today it's getting more and more complicated because it's getting harder and harder to figure out what's really causing the problem. It could be that the component that you're working with is just fine, but it happens to use other components inside of it that have caused some problems. **Site 1** Sometimes we're driven by constraints of time to market entry and clients' options of what they choose to do. It may initially be quicker just to write a new component as opposed to changing existing components to be
	Knowledge sharing	**Positive** Our company has several mediums of exchanges for us to get the most expertise and I think it takes pride in that. I mean we have a knowledge exchange, we can go into our databases to search for previous queries, you can get answers from all over the world. **Site 1** They were pretty open and honest about why they did things certain ways and also what problems there were with what they had done. **Site 3** **Negative** When you are busy, you can't really afford to spend time sharing what you know, you got so many other things going on. **Site 1** When crossing boundaries it was sometimes hard to get people to exchange information. **Site 3**

Appendix C: Identified cues for the constructs and indicators

Construct	Indicator	Example Cues
Strategy Change	Administrative guidelines	Standards, policies, rules, guidelines
	Resources	Allocate staff, allocate people, allocate time, allocate resources, allocate money
	Scope of change	Iterative, evolutionary, incremental, radical, major change.
	Education and training	Training programs, staff development, mentoring, education.
	Change agents	Reuse expert's role, authority, power.
Process Change	Structured implementation methods	Methodologies, development methods, development processes
	Tools	Repositories, libraries, tools, reuse technology, component development tools
	Process measurement	Reuse metrics, measurement
	Team-based structure	Communication, communicating tools, meetings, teamwork, collective achievement
Culture Change	Individual attitudes towards reuse	easy to create assets, right technology, easy to implement, easy to understand assets, easy to integrate assets, not invented here
	Knowledge sharing	Share knowledge, share experience, hoard, willingly exchange knowledge

About the authors

Karma Sherif is an Assistant Professor at the Rawls College of Business at Texas Tech University. She received her PhD degree from Texas A&M University. Her research focuses on barriers to the adoption of Software Reuse and knowledge management systems. Her work has appeared in Decision Support Systems, and Information and Management.

Nirup M. Menon is an assistant professor of management information systems at the University of Texas at Dallas. His research interests include exploring information systems management (MIS)-related phenomena using economics as the theoretical foundation, information technology productivity, impact of enterprise resource planning (ERP) systems, and Internet commerce facilitators such as trust, privacy, and cyberinsurance. He teaches ERP, data mining and data warehousing courses at UTD.

His articles have been published in Information Systems Research, Management Science, and Journal of MIS among others. He was previously an assistant professor at Texas Tech University. He received his PhD in MIS from the University of Arizona, Tucson. He is an associate editor of Information Systems Research and is a member of the IEEE, ACM, and INFORMS.

Journal of the Association for Information Systems
ISSN: 1536-9323