MIS Quarterly

RESEARCH ARTICLE

MANAGING PEER-TO-PEER CONFLICTS IN DISRUPTIVE INFORMATION TECHNOLOGY INNOVATIONS: THE CASE OF SOFTWARE REUSE[1]

By: **Karma Sherif**
Rawls College of Business
Texas Tech University
Lubbock, TX 79409-2101
U.S.A.
Karma.sherif@ttu.edu

Robert W. Zmud
Michael F. Price College of Business
University of Oklahoma
307 W. Brooks, Room 307E
Norman, OK 73019-4006
U.S.A.
rzmud@ou.edu

Glenn J. Browne
Rawls College of Business
Texas Tech University
Lubbock, TX 79409-2101
U.S.A.
glenn.browne@ttu.edu

Abstract

We examine the case of software reuse as a disruptive information technology innovation (i.e., one that requires changes in the architecture of work processes) in software development organizations. Using theories of conflict, coordination, and learning, we develop a model to explain peer-to-peer conflicts that are likely to accompany the introduction of disruptive technologies and how appropriately devised managerial interventions (e.g., coordination mechanisms and organizational learning practices) can lessen these conflicts. A study of software reuse programs in four organizations was conducted to assess the validity of the model. Qualitative and quantitative analyses of the data obtained showed that companies that had implemented such managerial interventions experienced greater success with their software reuse programs. Implications for theory and practice are discussed.

Keywords: Disruptive IT innovations, software reuse, goal conflict, coordination mechanisms, organizational learning

> Software reuse in general is like motherhood and apple pie: no one is against it. When you start talking about specifics…reuse philosophy becomes very difficult to implement.
>
> *An Application Developer*

Introduction

The challenges associated with adoption and use of information technology have long been a concern of scholars (Swanson 1994; Zmud 1982). Prior research has focused mainly on the management of vertical relationships to confront these challenges, striving to understand the antecedents

[1]Ron Weber was the accepting senior editor for this paper. Lars Mathiassen was the associate editor. Jeremy Rose served as a reviewer. The other reviewers chose to remain anonymous

to users' adoption and use of IT (Davis et al. 1989) and prescribing managerial interventions to facilitate adoption and use (Boynton et al. 1994). Often, however, an IT innovation significantly changes the architecture of work practices, disrupting peer-to-peer (horizontal) relationships that are critical to successful adoption and use of IT innovations. This study examines how management might intervene to maintain horizontal relationships, thus ensuring peer-to-peer cooperation in support of the IT innovation.

A disruptive innovation is a novel idea or behavior that, when introduced into organizational settings, causes dramatic changes in the architecture of work processes. It is usually accompanied by complementary administrative innovations (Abernathy and Clark 1985; Swanson 1994; Zmud 1984). Stakeholders who perceive their roles, tasks, and utility to be empowered by the innovation usually support its adoption. Stakeholders who perceive their roles, tasks, and utility as diminished usually resist its adoption (Orlikowski 1993). When coordination between conflicted peers is crucial in appropriating the potential benefits of the disruptive innovation, management must anticipate the likely goal conflicts and intervene in ways to resolve or at least reduce this divergence in stakeholder perspectives. Thus, in studying disruptive IT innovations, it is necessary to focus on both the cognitions of the peers expected to apply the technology (Lyytinen and Rose 2003) and the management strategies applied to direct and align peers' behavior (Kwon and Zmud 1987; Purvis et al. 2001).

Disruptive IT innovations in today's organizational settings are common events, and, if anything, the incidence of such implementations is accelerating. An increasingly prominent form of disruptive innovation is the introduction of new IT-enabled platforms (Purvis et al. 2001) or tool-kits (von Hippel and Katz 2002) from which work processes or activities are orchestrated. Such platform-hosted work process innovations are perhaps best interpreted as architectural innovations (Henderson and Clark 1990; Henderson and Cockburn 1994), as they induce change to both work process components and the linkages among these components.

In the present research, the introduction of a software reuse initiative is viewed as a disruptive IT innovation that induces significant changes in the work process architectures used by adopting software development groups. This is accomplished by building a platform consisting of component architectures, software assets, and related functionality. Reuse changes the process of development from the analysis, design, and implementation of customized software to the development of reusable components applied to develop a stream of applications (Basset 1997).

We focus on the effect of the change in software development processes associated with two vital, but likely conflicted, peers: asset creators and asset users (i.e., application developers). The complexities experienced by asset creators in developing flexible, error-free reusable components and the pressures of time-to-market applications and budget overruns experienced by asset users are likely to result in goal conflicts between these two peer groups (Banker and Kemerer 1992). Using conflict and coordination theories, we explain the peer-to-peer conflicts that accompany the introduction of software reuse and identify formal and informal strategies for managing these conflicts. The potential for organizational learning to reduce goal conflicts is also incorporated within the proposed research model. Stated simply, our argument is that effective coordination and learning structures enable a software group's members to evolve work norms and practices consistent with the new work processes, thereby resolving much conflict that may otherwise occur (Jehn and Mannix 2001).

These ideas converge in the following three research questions:

- What is the nature of the goal conflicts that arise between interdependent but conflicted peers associated with the introduction of disruptive IT innovations?
- What is the nature of the formal and informal coordination mechanisms to be applied in managing goal conflicts such that positive outcomes can be attained from the adoption of disruptive IT innovations?
- What is the nature of the organizational learning mechanisms to be applied in managing goal conflicts such that positive outcomes can be attained from the adoption of disruptive IT innovations?

Given limited understanding of the effect of disruptive innovations on peer-to-peer relationships, in-depth case studies of four reuse programs in three different industries were conducted. We follow theoretical and literal replication logic in the selection of cases to examine how well the experiences of actors associated with successful and unsuccessful reuse programs can be generalized to the theoretical constructs of our model. Both qualitative and quantitative analyses are employed. The next section develops the study's research model, positioning it in the context of prior research. Next, the research strategy and design are described, the data analyses and findings are reported, and the implications of these findings are discussed.

Development of the Research Model

A disruptive innovation "deviates radically from existing product lines by incorporating novel and unprecedented architectural principles" (Lyytinen and Rose 2003, p. 560). When a disruptive innovation enables a platform through which an organization's members execute specific work processes, the architecture of these work processes (i.e., the underlying composition of work components and linkages between the components) is likely to change dramatically. Evolving the knowledge and relationship structures fundamental for these newly instituted work processes is complicated, as complementary changes to communication channels, information filters, and problem solving strategies are required (Henderson and Clark 1990)

Software reuse is a process innovation that strives to maximize the use of reusable components in building a stream of software applications within a specific domain (Prieto-Diaz 1993; Tracz 1987). In essence, reuse shifts the focus of a software development group from developing relatively independent software applications to populating a repository with architecturally related software components to be applied in fabricating a stream of software applications (Basset 1997; Fichman and Kemerer 2001; STARS 1996). Asset creators (a new work role) participate in an upstream process (a new work process) to populate the repository, while asset users are involved in a downstream process to use the repository's software assets in creating solutions for clients. A software reuse program thus transforms the software group's "throughput technology" (Ettlie and Reza 1992; Hatch and Mowery 1998) in the course of introducing new development processes, such as domain analysis and component-based design (Kim and Stohr 1998; Prieto-Diaz 1993). While prior research has identified several technical and cultural conflicts associated with the adoption of reuse (Banker and Kaufman 1990; Gomaa and Kerschberg 1995; Kim and Stohr 1998, Ravichandran 1999; Sherif and Menon 2004; Sherif and Vinze 2003), little prior research has focused on peer-to-peer conflicts likely to arise between asset creators and asset users and associated administrative changes introduced to coordinate efforts and align peer behaviors. Given both the substantial investment required in developing technological and human capital infrastructures and the necessity to establish a mutually reinforcing choreography addressing the interdependencies between the creation of reusable assets and the development of component-based applications (Hooper and Chester 1991; Poulin 1997; Sherif and Vinze 2003), the goals of asset creators and asset users need to be aligned.

In the early phases of a software reuse program (i.e., when the repository is sparsely populated), it is unlikely that such choreography is in place (Ravichandran 1999; Vitharana 2003). Asset creators are invariably enthusiastic about reuse. They wish to exploit the potential of each new application development project fully to enrich their understanding of the software domain, to evolve the domain architectures, and to populate the repository with maximally reusable software assets. Asset users, however, remain compelled to deliver solutions that fulfill their clients' requirements in a timely, cost-efficient manner (Banker and Kemerer 1992). At best, asset users are likely to view the reuse initiative as an obstacle to goal achievement. Over the long-term, asset users should be able to appropriate considerable value from the reuse program, but only if reusable assets are (1) available and easily located, (2) sufficiently adaptable to meet clients' needs, and (3) developed to standards such that the components can be integrated easily with other reusable assets as well as with custom-built components (Vitharana 2003). In the short-term, however, few reusable assets are likely available, those that are available are unlikely to be adapted easily to a specific client's needs, and, at best, an immature set of asset standards is likely to exist. It is thus very likely that in the early stages of a software reuse program, conflicting goals will surface between asset creators and asset users (Austin 2001; Banker and Kaufman 1992; Fichman and Kemerer 2001). Without appropriate interventions by a software group's management to address goal conflicts, the reuse initiative is unlikely to be sustainable. The research model shown in Figure 1 identifies two such interventions (i.e., coordination and organizational learning structures) and posits that their use will reduce the extent of goal conflict between asset creators and asset users, leading to more favorable outcomes for a reuse program. The remainder of this section will develop this research model, moving right to left.

Reuse Program Positive Outcomes

The benefits realized from reusability on the cost, delivery time, and quality of software applications as compared to the investment cost attributed to the reuse program define the value of such a program (Poulin 1997). By leveraging a repository of reusable components, reuse advocates claim that reductions will be experienced in the cost of and time to develop applications, as only a small portion of the code involved with an application will be built from scratch. Reuse is also expected to improve the quality of software because reusable assets—having gone through multiple iterations of testing and quality enhancements—are likely to have fewer errors than freshly developed code. Thus, stakeholders associated with successful software reuse programs, relative to those associated with less successful reuse programs, should favorably perceive programmatic impacts on development costs, timeliness, and quality.

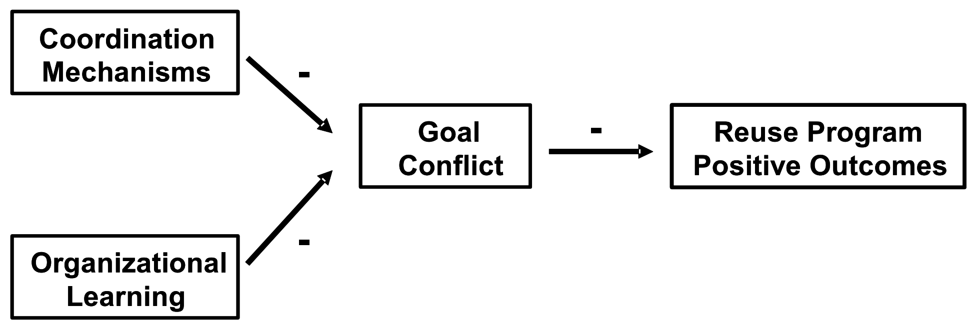

Figure 1. The Study's Research Model

The Influence of Goal Conflicts on Reuse Project Outcomes

Goal theorists (Austin and Vancouver 1998) stress the need for interpreting outcomes of organizational processes as a function of the efforts by actors to achieve their individual goals. Individuals usually strive to attain multiple goals that are structured hierarchically (Hyland 1988), with abstract high-level goals being attained by accomplishing more tangible low-level goals. One implication of such goal hierarchies is the potential for goal conflicts (Abraham and Sheeran 2003). In particular, while people involved with an organizational process may share a common high-level goal, their pursuit of specific low-level goals may become conflicted (Karoly 1998).

Thus, a major source of the goal conflicts arising during the adoption of disruptive innovations is posited to be process conflicts, that is,

> conflict about how task accomplishment should proceed in the work unit, who is responsible for what, and how things should be delegated. Process conflict includes disagreement about assignment of duties or resources. (Jehn 1997, p. 540)

While all stakeholders might agree regarding the expected long-term benefits of a disruptive process innovation, the adoption of the innovation may constrain the ability of some stakeholders to attain their short-term goals. When the simultaneous achievement of multiple goals is infeasible, goal conflicts inevitably arise (Carver and Sheier 1998). The presence of goal conflicts, along with the difficulties of attaining goals, likely results in a lack of commitment, a decline in expended effort, and a drop in performance (Locke et al. 1988).

This seems to be the case with software reuse. At the onset of a reuse program, the time and cost associated with the changes in the components of development processes and the linkage between the components are major sources of potential conflict between asset creators and asset users (Fichman and Kemerer 2001; Jaber et al. 2000).

Software reuse programs are expected to increase the *time* involved in developing software solutions in the absence of a critical mass of reusable components, as each software solution project must focus both on meeting clients' needs and creating reusable components. The task of developing software components for reuse is considered more challenging than the task of developing software components of a typical application (Rothenberger et al. 2003). This is due to the need to construct software assets relevant to an entire software domain rather than a single, well-defined aspect of that domain, and the need for those assets to continue to be relevant in the future (Vitharana 2003). Even after the component repository has begun to be populated, new solution development processes must be designed, learned, and practiced. The reuse archive must be searched to locate components that might be useful for an application (Vitharana et al. 2003). Once identified, the concept, content, and context of each of these possibly salient components must be understood (Prieto-Diaz 1993). Next, explicit decisions must be made regarding each element of a software design as to whether required functionality can be met via a reusable component or must be customized. Finally, a lack of mature standards for designing interoperable reusable assets may also make it difficult to integrate components from different sources in fabricating a software solution.

The *cost* (i.e., expended resources) of developing reusable assets during an application's development has been described as exorbitant, with no significant short-term returns (Frakes and Isoda 1994; Poulin 1997). For an asset user having to interact with a client and meet his or her needs, creation of reusable components is likely to be viewed as an excess cost, as the expended resources are not directly leveraged by the client in the solution being developed (Griss et al. 1994; Hooper and Chester 1991). Instead, reusable components built during a specific development effort are leveraged only by subsequent projects (Banker and Kemerer 1992; Rothenberger et al. 2003), which may or may not involve the same client.

In summary, the higher the incidence of conflict between asset creators and asset users, the less likely it is that an effective upstream process focused on asset creation has been developed and aligned with a redesigned downstream process of asset use in fabricating client solutions. Further, such goal conflict is likely to lead to a lack of commitment toward reuse programs by both asset creators and asset users. Consequently, software reuse programs characterized by high levels of conflict between asset creators and asset users are likely to be less successful.

Coordination Mechanisms

Research examining software reuse has focused consistently on facilitating the vertical relationships between top management and stakeholders impacted by a reuse initiative (Banker and Kauffman 1990; Kim and Stohr 1998; Rine and Sonnemann 1998; Rothenberger et al. 2003), while largely neglecting peer-to-peer horizontal-level coordination of asset creators and asset users. Given the interdependency between upstream and downstream agents, along with the potential for agents to behave opportunistically (Hoskisson and Hitt 1988), coordination mechanisms targeted at these horizontal relations are critical for controlling and integrating work activities of these potentially conflicted agents (Montoya-Weiss et al. 2001). Such coordination mechanisms serve to increase the likelihood that each agent makes decisions consistent with the overall welfare of the organization (Jensen and Meckling 1976).

By coordination we mean "the mode of linking together different parts of an organization to accomplish a set of collective tasks" (Andres and Zmud 2001, p. 45). Coordination mechanisms are particularly suited for work environments characterized by goal conflict, task interdependence, and task uncertainty (Andres and Zmud 2002; Choudhury and Sabherwal 2003; Kirsch 1997). Mechanisms typically advocated include monitoring work processes required in carrying out interdependent tasks and rewarding cooperative behaviors (Itoh 1991, Sinclair-Desgagné 1999) along with facilitating open communication to promote sharing of work-related information (Andres and Zmud 2001; Montoya-Weiss et al. 2001).

Monitoring an agent's efforts to ensure that the agent makes decisions consistent with the overall welfare of the organization (Jensen and Meckling 1976) is expected to reduce goal conflict among interdependent agents. In a reuse context, such monitoring needs to be directed at two quite different sets of behaviors: the upstream process to construct software assets and the downstream process of fabricating solutions for clients. Reuse initiatives can increase development time and cost, but this additional time and cost is of little benefit to the specific application under development. Thus, monitoring the behaviors of both asset creators and asset users as well as measuring the effect of using reusable assets in application development are strongly advocated (Poulin 1997; Rothenberger and Hershauer 1999).

Well-constituted ***reward*** mechanisms can also prove to be an effective form of agents' self-regulation when tied to the behavior or outcomes desired (Eisenhardt 1985, 1989; Levinthal 1988). Reward mechanisms foster cooperation between multiple agents by aligning their goals with those of the organization (Nilakant and Rao 1994, Sikora and Shaw 1998). In particular, incentive schemes in which (1) collective outcomes rather than individual outcomes are rewarded (Sharma and Yetton 2003), (2) upstream agent rewards are contingent on the work outcomes of the downstream agent (Itoh 1991, Sinclair-Desgagné 1999), and (3) information sharing is rewarded (Sharma and Yetton 2003) are likely to prove most effective in reducing goal conflicts among interdependent agents. Nonetheless, despite beliefs that appropriately designed incentive structures induce asset users to increase their use of reusable assets (Fishman and Kemerer 2001; Ravichandran 1999), no empirical evidence exists of a positive relationship between reuse-based rewards and the outcomes of a reuse program (Frakes and Fox 1995).

Communication processes serve as a third coordination mechanism for managing conflict. Such processes manage conflict by enabling interdependent agents to become knowledgeable about each other's work roles as well as the status of assigned work tasks. In this way, their efforts can be coordinated to produce the final output (Crowston 1997). Moreover, alternative perspectives and potential disagreements can be surfaced. In the case of software reuse, communication among asset creators and asset users needs to be reciprocal. Asset users, being closer to the client and hence to application domains, must be involved in the analysis and design of domain architectures and reusable assets. On the other hand, asset creators best understand the existence and internal structure of reusable assets and how these might best be adapted to fit solutions being designed for clients. Thus, they must be involved in the design of client solutions.

Organizational Learning

At the early stages of a software reuse program, designing an effective set of coordination mechanisms may be difficult, especially if the software group lacks sufficient knowledge and understanding of "desired" processes, behaviors, and outcomes. Organizations learn when they encode inferences from past experiences into conceptual frames and eventually into routines that guide behavior (Arrow 1962). As experience is accumulated, organizations grow relevant knowledge stocks. Hence, they can positively impact the efficiency and effectiveness of their production functions (Argote 1993). Among the learning processes that determine effectiveness are conceptual and operational learning (Mukherjee et al. 1998).

In an organizational setting, ***conceptual learning*** entails specification of processes required to perform tasks (Eisenhardt 1989). In the context of software reuse, conceptual learning involves (1) defining standards and guidelines associated with development and integration of reusable assets (STARS 1996), (2) incorporating reuse in all phases of application development processes (Gomaa and Kerschberg 1995), and (3) developing domain-specific architectures (Basset 1997). As software groups gain experience with reuse across domains, they abstract lessons learned into formal standards and guidelines. Integration of these standards and guidelines into development processes and creation of domain-specific architectural frameworks that describe the backbone structure of all applications within a domain (Gomaa and Kerschberg 1995) facilitate development of generic solutions for recurring tasks (Nelson and Winter 1982). Thus, software groups with formal standards and guidelines, reuse-friendly development methodologies, and domain architectures should exhibit reduced levels of goal conflict among asset creators and asset users.

Operational learning involves change in the organization's collective understanding (Dusya and Crossan 2004) through the exploitation of lessons learned and their assimilation within work processes (Mukherjee et al. 1998). As the organization faces new challenges, the need may arise to evolve the already established standards and guidelines to account for novel experiences. A key aspect of operational learning is measuring the effectiveness of development processes in solving new problems and determining ways to restructure standards and guidelines to realize the desired outcome. In a software reuse setting, operational learning measures second-degree learning manifested in the integration of lessons learned within the conceptual frames and evolving standards for developing and integrating reusable assets, domain architectures, and development processes (Poulin 1997; Rothenberger and Hershauer 1999).

The Interaction Between Coordination Mechanisms and Learning

Interactions between coordination mechanisms and organizational learning enable organizations to manage peer-to-peer conflicts surrounding adoption of disruptive innovations. On one hand, organizational learning activities facilitate development and evolution of an effective set of coordination mechanisms. As organizations specify standard processes for carrying out tasks, a frame of references is set (and communicated) for monitoring activities and rewarding desired behaviors and outcomes. On the other hand, deployment of coordination mechanisms, particularly monitoring and communicating, are critical for operational learning (Pich et al. 2002). As an organization monitors the implementation of standards and guidelines, it is likely to encounter new situations not accounted for previously. Communicating these new experiences triggers operational learning and then conceptual learning, fostering restructuring of standards and guidelines and subsequent adaptation of coordination structures and outcome expectations.

In summary, in the early stages of software reuse programs, asset creators are expected to focus on producing domain architectures and populating the reuse repository with assets that conform to those architectures. Asset users, on the other hand, are expected to focus on building software solutions to meet client needs in a timely and cost-effective manner. As a result of the different goals of asset creators and asset users, changes in the architecture of development processes will invariably cause conflicts between the two peers. However, as managers intervene by introducing effective coordination mechanisms (monitoring, rewarding, and communicating) and deploying effective organizational learning practices (conceptual and operational) to align the behaviors of these interdependent peers, the incidence of conflict among asset creators and asset users should subside.

Research Methodology

Given the intent of this study of capturing a subtle pattern of human interaction, the methodology followed was a multiple case study approach (Benbasat et al. 1987; Miles and Huberman 1984; Yin 1994). Such a research strategy seemed most likely to surface goal conflicts among asset creators and asset users.

We follow the positivist perspective in developing and then assessing a research model, examining how well it corresponds with the experiences of actors associated with reuse programs in software developing organizations. We restrict our data collection and analysis to our theoretical foundation and the *a priori* set of constructs in the research model. We do not attempt to reconstruct the meaning developers create and attach to software reuse and their behavior toward the technology as in interpretive studies (Dubé and Paré 2003; Klein and Meyers 1999). Thus, the units of analysis for this study are the discrete statements, extracted from interview data, regarding reuse outcomes, goal conflicts, coordination mechanisms, and organizational learning by individuals actively involved with software reuse initiatives in selected software development groups. Both qualitative and quantitative analyses are employed to validate the model and enrich our understanding of disruptive innovations.

Site Selection and Data Collection

In selecting sites for data collection, a letter was sent to 53 corporate members of an advisory board for the school of business of a southern U.S. university soliciting participation in the study. Thirty-five software groups were identified as having implemented a software reuse program. A preliminary interview was conducted with at least one individual from each organization to ensure that the reuse program was formal and structured. Four selection criteria were then applied:

- The reuse program had specific goals and objectives.
- The reuse program was in an early stage of its deployment, such that the reusable asset repository was in the process of being populated.
- The asset creator role had been formally defined and instituted.
- A clear distinction existed between the roles of asset creator and asset user.

We also followed a combination of literal and theoretical replication strategies to improve the external validity of our findings. Theoretical replication refers to a multiple-case selection strategy in which the cases vary in terms of expected outcomes (Yin 1994). Accordingly, we selected two sites (ITA Consulting and ITB Consulting) that reported successful reuse programs and two others (Global Telecom and Oil & Gas Co.) that reported unsuccessful reuse programs. Literal replication, on the other hand, requires selecting cases that are similar in certain characteristics and thus lead the researcher to expect similar results (Yin 1994). All four sites represented large, multi-national organizations with large IT development budgets and strong IT management reputations. We expected the two successful companies to show convergent evidence of the theoretical conditions posed in our model as requisite for a positive reuse outcome (literal replication). The lack of the same conditions at the two unsuccessful sites was expected to explain failure (theoretical replication). The cases were purposefully selected across different industries so the findings were not industry specific.

ITA Consulting is an IT services company that focuses specifically on client/server and advanced technologies for the energy, communications, and financial services industries as well as government agencies. Five individuals were interviewed at ITA Consulting, and all were involved with a specific project being undertaken for a major telecommunications firm.

ITB Consulting is a major software consulting organization. Here, interviewees were from the company's Energy Solution Group, which develops gas accounting systems for energy firms. Five reuse stakeholders were interviewed; all were involved with a common set of projects.

Global Telecom is one of the world's major communications and information services companies. Interviewees were all with the company's Customer Billing Services (CBS) unit, which provides billing services and customer-care applications for residential customers. Eight reuse stakeholders were interviewed. At the time of the study, these interviewees were involved with a number of overlapping projects.

Oil & Gas is an oil and gas company with worldwide operations. Interviewees were associated with the company's Production and Operation Management (POM) group, which delivers IT solutions for refineries or chemical plants. Twelve reuse stakeholders were interviewed. All were collaborating on a common set of projects.

The analysis strategy has two components, one quantitative and one qualitative. The set of interviewee statements was reduced for the quantitative analysis. Given the focus on surfaced conflict among asset creators and asset users, only the interviews within a site with individuals serving these two specific roles on a common set of projects were analyzed quantitatively. Thus, we eliminated from the data analysis the

Table 1. Demographic Distribution of Informants at the Four Sites

Site	Job Title	Years of Experience	Gender
ITA	Reuse Expert/ Director of the IT Unit	16	Male
	Systems Architect Senior Manager/ Asset creator	12 8	Male Male
	Asset user 1 Asset user 2	6 3	Male Male
ITB	Reuse Expert/ Director of the IT Unit	14	Male
	Systems Architect Senior Manager/ Asset creator	10 10	Male Male
	Asset user 1 Asset user 2	4 1	Male Female
Global Telecom	Reuse Expert/ Senior Manager	20	Female
	Systems Architect Asset creator	13 10	Male Male
	Asset user 1 Asset user 2	5 4	Male Male
Oil & Gas Co	Reuse Expert 1 Reuse Expert 2 Reuse Expert 3	25 15 12	Male Male Female
	Systems Architect Senior Manager Asset creator	10 10 7	Male Male Male
	Asset user 1 Asset user 2 Asset user 3 Asset user 4	5 5 4 6	Male Male Male Female

interviews of the project manager from ITA Consulting, the chairman of the reuse working group, the project manger and the manager of the CBS unit at Global Telecom, and the project and senior manager at Oil & Gas. This culling resulted in the transcript data set for the quantitative analysis consisting of statements from five interviewees each from ITA Consulting, ITB Consulting, and Global Telecom, and 10 interviewees from Oil & Gas. However, the total set of interviewee statements, including those eliminated from the quantitative analysis, was used in the qualitative analysis. Demographic data of the respondents at each site appear in Table 1.

In an effort to avoid biasing responses, data were collected via a semi-structured interview guide using a common set of general, open-ended questions. The questions were worded so as to elicit as many perceptions as possible regarding the reuse initiative at each of the four companies. These questions were:

- Explain your current role within the organization and the type of tasks you are charged with.
- What does software reuse mean to you?
- Do you believe the technology is beneficial?
- Do you believe the technology is easy to implement?
- Do you believe reuse is successful in your organization?
- What factors contribute to the success of reuse?
- What factors constrain the adoption of reuse?

Table 2. Description of Coded Concepts

Construct	Subcategory	Description
Goal Conflict	Costs	Conflict surfaced around an application development project's budget.
	Time	Conflict surfaced around an application development project's elapsed time.
Coordination Mechanisms	Monitoring	Monitoring the development and integration of reusable assets and measuring their effect on quality, cost, and time of application development.
	Rewarding	Providing rewards to promote adherence to reuse standards and guidelines and to promote the integration of reusable assets with applications being developed.
	Communicating	Developing formal and informal communication channels to facilitate the flow of information between asset creators and asset users.
Organizational Learning	Conceptual Learning	Developing standards and guidelines to create and integrate assets into applications being developed.
	Operational Learning	Exploitation of lessons learned in evolving reusable assets, domain architectures, and standards and guidelines.
Positive Reuse Outcomes	Outcome of a Reuse Program	Positive impacts of reuse on quality, cost, and time of application development, on customer satisfaction, and on profitability.

Coding of Interview Transcripts for the Quantitative Analysis

Transcriptions of the recorded interviews were performed to facilitate analysis. In total, over 500 pages of transcribed interview data were produced. A first round of content coding was performed by one of the coauthors. Krippendorff's (1980) approach to content analysis was followed, in which words from the interviews were first assigned to one of three high-level categories (goal conflict, coordination mechanisms, and organizational learning) and then further categorized within the following categories: for goal conflict, conflict on costs and time; for coordination, monitoring, rewarding, and communicating; and, for organizational learning, conceptual and operational. In addition, interviewee comments were also searched for statements linking the reuse program to favorable application development or financial outcomes. Table 2 provides descriptions of the constructs as well as their subcategories.

Cues for data categorization were identified. Sample words and statements were tabulated for each theoretical code. In the second round, we recoded the data using NUD*IST software for content analysis. Counts provided by the software represented the frequency for each of our theoretical categories. A second researcher, who was trained in content analysis but blind to the research hypotheses, also coded the data. This independent researcher was provided with only the description and sample cues for our theoretical categories (Table 2). The inter-rater reliability for the two coders was slightly above 90 percent.

Results

In answering the research questions, the data first were examined quantitatively in light of the relationships proposed in the research model (Figure 1). Counts of statements made under each construct subcategory were summed (Table 3) across interviewees, accounting for the number of interviewees for each site. Table 4 reports the proportion of coded comments for each construct category for each site. The pattern similarity of Table 3 and Table 4 indicates that the differences observed between the successful and unsuccessful case sites is not biased by the number of statements made at each site.

If the posited relationships were to hold, successful reuse programs would report a higher incidence of positive outcomes, a lower incidence of goal conflict, a higher incidence of enacted coordination mechanisms, and a higher incidence of enacted organizational learning. Visual inspection of Table 3 indicates that the counts conform to these expectations. The χ^2 results provided in Table 5 show that these differences are significant.

Qualitative analyses of the data were also conducted. A summary was generated for each case highlighting evidence

Table 3. Construct Coding by Company[a]

	Total Comments	AVG	Total Comments	AVG	Total Comments	AVG	Total Comments	AVG
Positive Reuse Outcomes	100	25	140	28	28	5.6	19	1.9
Goal Conflict								
Conflict on Time	14	3.5	13	2.6	90	18	110	11
Conflict on Costs	11	2.75	15	3	21	4.2	58	5.8
Total	25	6.25	28	5.6	111	22.2	168	16.8
Coordination Mechanisms								
Monitoring	45	11.25	56	11.2	19	3.8	16	1.6
Rewarding	35	8.75	51	10.2	12	2.4	27	2.7
Communicating	44	11	26	5.2	7	1.4	4	0.4
Total	124	31	133	26.6	38	7.6	47	4.7
Organizational Learning								
Conceptual Learning	40	10	63	12.6	7	1.4	3	0.3
Operational Learning	7	1.75	12	2.4	2	0.4	0	0
Total	47	11.75	75	15	9	1.8	3	0.3
Total Number of Comments	296	74	376	75.2	186	37.2	237	23.7

[a]AVG was calculated by dividing a company's total comments by the number of interviewees.

Table 4. Proportion of Statements by Construct

	ITA Consulting	ITB Consulting	Global Telecom	Oil & Gas
Positive Reuse Outcomes	0.34	0.37	0.15	0.08
Goal Conflict	0.08	0.07	0.60	0.71
Coordination Mechanisms	0.42	0.35	0.20	0.20
Organizational Learning	0.16	0.20	0.05	0.01

Table 5. Results of χ^2 Analysis

	Positive Reuse Outcomes	Goal Conflict		Coordination Mechanisms			Organizational Learning		
		Time	Cost	Monitor	Reward	Comm	Concept Learning	Operation Learning	Total
Successful	240	27	26	101	86	70	103	19	672
Unsuccessful	47	200	79	35	39	11	10	2	423
Chi-Square	129.787	131.846	26.752	32.029	17.672	42.975	76.54	13.762	
p-value	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	

of goal conflicts, coordination mechanisms, and organizational learning, as well as causal relationships between constructs. Within- and cross-case comparisons were conducted to identify and match patterns. Evidence of literal replication (convergent results within similar sites) and theoretical replication (divergent results across different sites—successful and unsuccessful) were tabulated. Support for our replication logic is provided below.

Reuse Program Outcomes

Examining the comments regarding the effect of reuse on the budget, development time, and quality of software applications, significant differences can be seen in the views of program outcomes between the more successful and the less successful sites. Strikingly, developers at ITA Consulting and ITB Consulting stressed that a primary benefit of the availability of reusable components was that these components enabled the early demonstration of "solutions" to clients, which clients perceived as exhibiting competence in delivering high-quality software on time.

> The way we convince our client to implement reusable solutions is we show them that we get the same results. We take a month's worth of their data, and we put it into our system and set it up and we compare them to their reports. When the client can see that, "Hey, I'm getting the right answers here," then that's pretty easy to have buy-in from them, then it's just a matter of showing them how to use the system. The client's generally real happy and doesn't mind doing things our way because we're coming to the same conclusion, and generally it's easier for them to use.
>
> *Senior Manager, ITB Consulting*

Furthermore, both asset creators and asset users expressed certainty they would continue to achieve benefits from future reuse because of the gains realized.

> We are more likely to continue to succeed in the future. We've definitely seen reductions in the amount of time. If we compare the amount of effort to develop the initial releases of our application to the amount of effort required to develop the second releases, less, less effort for the second release, and that was despite the fact that it was comparable in the actual amount of functionality that was delivered to the users. With reuse we have a much better chance of estimating and mitigating risk because, again, we've done it before, we know the problem space, some stuff's pre-built, but we also know what that effort's going to entail.
>
> *Asset User 2, ITA Consulting*

Different views surfaced at Global Telecom and Oil & Gas, where asset users expressed skepticism about promises made by reuse advocates. With the exception of a few anecdotal experiences, most reuse development projects were unsuccessful. Neither company experienced a reduction in cost or time attributed to the quality and availability of reusable assets.

> If you look at the trend, you would see that there's not an overwhelming amount of reuse. We're putting a lot of effort into it which we won't get paid out till, you know, two or three years down the road….My own perspective, I think that software reuse is probably overemphasized in life and in the industry today.
>
> *Senior Manager, Oil & Gas*

> Our biggest problem is that we do not do enough of the upfront work. We're not doing a lot of the modeling correctly, we're not doing a lot of the modeling period. We don't have a lot of discipline around that right now….components were not built to be reusable, so for someone to introduce a new function that has reusable features, it creates more work.
>
> *Asset Creator, Global Telecom*

Goal Conflict

It also seems quite clear that goal conflicts are, as posited, inherent aspects of a disruptive innovation such as software reuse. Reducing the cost and time of development via reuse requires a change in the architecture of development processes, which shifts resources from customized development to domain analysis and reusable component design, thereby creating an inherent conflict.

> That's just the nature of the beast; it takes more time to design a reusable component because you are thinking of more than just the immediate problem. It takes more time and your best developers to think out of the box and think in the future sense, what might happen in the future, what are some of the scenarios, how it could be used generically by other places and also meet the current requirement that we have. So it's usually a little bit of a longer process to design for reuse.
>
> *Asset Creator, ITA Consulting*

> Individual projects are so tightly tied to meeting schedules and deadlines that it's difficult to get them to look beyond their own project's needs, and it's difficult to get them to put in any effort whatsoever to develop a component that might be reusable outside their group. Developing components that could be used on other projects at an undetermined date down the road will always be scrutinized when we're in an environment that has to deliver a product to the field so fast. It's very difficult. There's no motivation for it.
>
> *Asset User 4, Oil & Gas*

The interviewees recognized that the availability of reusable assets does not guarantee a reduction in the cost and time of application development but instead can increase risk borne by both asset creators and asset users.

> The problem with developing a component that can be reused up front is that you really don't know how it's going to be used in other applications; you don't really have the requirements for the other applications. You always run the risk of not knowing exactly what the thing is doing behind the scenes…. *I think it's a risk well worth taking* [emphasis added].
>
> *Asset Creator, ITB Consulting*

A key difference observed between the successful and unsuccessful sites was that developers at ITA Consulting and ITB Consulting were able to align their goals—that is, reduce conflict inherent in the upstream and downstream processes—and saw reuse as a viable method to achieve these joint goals. In particular, there was an emergence in these two successful sites of a shared vision regarding software reuse.

> To a large extent our ability to implement reuse depends on our culture and the willingness of the organization to invest in developing the infrastructure that's necessary to facilitate reuse….The organization is really geared up towards looking for existing components. You get people who understand the value.
>
> *Reuse Expert, ITA Consulting*

> We rely on the fact that reuse is a shared vision…. Reuse will make your life easier and the company earns more if you produce reusable components.
>
> *Asset User 1, ITB Consulting*

On the other hand, managers and reuse experts at Oil & Gas and Global Telecom realized that goal misalignment was a significant challenge to be overcome.

> There's not a perceived commitment to make reuse happen because we haven't had much success with it in the past. We are not willing to foot the short-term pains for the potential of long-term gains.
>
> *Senior Manager, Oil & Gas*

> It's very difficult to get people to adopt it. They lip service through it. They'll say, "Yes, we realize that it's important," but when it comes right down to dedicating people, that's when it gets difficult.
>
> *Reuse Expert, Global Telecom*

Coordination Mechanisms

It is evident from the data collected that ITA Consulting and ITB Consulting enacted multiple strategies to reduce goal conflicts among asset users and asset creators. Through close monitoring of solution development, they recognized and reacted to deviations from desired reuse behaviors.

> There is a pretty rigorous review process involved in development so reinventing is caught. If the staff person did not recognize that something should be leveraged or published as reusable, the supervisor should. We are able to catch the problem because we have structured a new process.
>
> *Reuse Expert, ITA Consulting*

They also monitored the development of reusable assets and made sure the assets were of high quality and could be integrated easily within applications.

> We started to build our architectures and infrastructure components to help us be more productive and develop in a more consistent way. We focus on how does one provide and create vertical content and become more valuable within an industry segment, a vertical segment…. We make sure that the assets are really best solutions, solutions that have flexibility built into them. We try to design in a manner that can let different people do different things with it without having to modify the code at all.
>
> *Reuse Expert, ITB Consulting*

Furthermore, impact evaluations of reuse on solution development made it easier for asset users to see the benefit of reuse and hence deem the associated risks to be justified.

> It used to take a week to build that functionality. That was about a year ago. Now with the architectures and the components, it makes it easier to build it in just about a day. That is kind of hard to pass

> unnoticed. The very first time we reused it, it saved us 40 to 45 percent of the time. Now it saves us 75 percent to 100 percent, somewhere in that range. So you reap the rewards fairly quickly.
>
> *Asset User 2, ITB Consulting*

While neither consulting company provided direct monetary rewards for reuse, both asset creators and asset users clearly felt strongly that their performance evaluations were tied directly to their adopting reuse-oriented behaviors.

> We have a pretty extensive evaluation process. I think it is pretty cool. The more reusable components they can create that other people use the better evaluation they get….that supports them in promotion….The evaluations are done based on what they've accomplished and kind of how well they have interacted with the application groups just to achieve those accomplishments.
>
> *Reuse Expert, ITB Consulting*

Coordination efforts established at the two consulting firms also focused on creating information flows between asset creators and asset users.

> We are sending out a quarterly letter of what they've developed here and an explanation of the components. We have a monthly meeting where we talk about components that are developed to make developers more aware of what things are out there. Whoever develops the component would then build a use case for it and everyone would have an example of what it does and how it does it. We have to have very good communication between the different reuse teams and our different application groups that are going to use or reuse a particular object or component, because without the proper communication, which I think is probably a key factor, we don't really know how the components are used internally. The developers are relaying back any changes to the software that we need to do to help them do their jobs.
>
> *Asset Creator, ITA Consulting*

Asset creators engaged in aggressive marketing of reusable assets to asset users and promoted success stories to management to reduce skepticism about reuse. Asset users provided periodic feedback on the quality and performance of reusable assets in the applications they had built. Communication also helped asset creators better understand the domain requirements and assisted the asset users in adapting reusable assets in new solution contexts. Both companies dispatched asset creators to individual projects to heighten awareness of reusable assets.

At Global Telecom and Oil & Gas, on the other hand, minimal monitoring and rewarding of reuse behaviors were observed. Reuse was left to opportunities rather than systematically linked via monitoring and rewarding to work practices associated with either the development of reusable assets or their integration into client solutions. At best, limited attention to communication was evident as well. Neither company supported active marketing and servicing of reusable assets, causing asset users to lose opportunities for reuse because they were unaware of an asset's existence or because they feared lack of support during development or after delivery.

> Communication has to be interjected by either organizational structure, or routine type sessions, or where maybe you get people together to talk about the different reusable pieces that they came up with instead of snippets here and there, which people don't seem to collect. So, basically, the communication is very null, unless you interject it through a role.
>
> *Reuse Expert 3, Oil &Gas*

Organizational Learning

A substantial difference in the level of organizational learning surfaced between the successful and unsuccessful case sites. At ITA Consulting and ITB Consulting, the software development group was structured along sector lines, enabling both companies to exploit for purposes of reuse the commonalities among solutions developed within a sector. This structure also allowed development of domain-specific skills involved in implementing reuse. Instead of opportunistically reusing assets, asset creators exploited sector commonalities within sector-specific architectures, providing the backbone design of applications within a domain. Standards and guidelines were then defined for creation and integration of assets. These were embedded within reengineered development methodologies.

> We've also put together some methodology and processes for how to model components, how to define the difference between what the component offerings and what the client already has, also how to model the internal functionality of a component. Our best practices are wrapped into our methodology….those best practices and lessons learned are worked back into the methodology and into the framework by the people that have learned.
>
> *Reuse Expert, ITA Consulting*

The sector-specific architectures, along with the newly devised methodologies established for the creation of reusable assets and development of client solutions, thus served to institutionalize work practices around the reuse program.

> The domain architecture piece, which is kind of generic across all our systems, is a common metaphor of how common functionality will work across all our products. It is a common design pattern. … Anything that we can push up into the domain architecture layer makes it easier for us to customize it for new clients….It facilitates a lot of the main thing that developers have to deal with.
>
> *Reuse Expert, ITB Consulting*

ITA Consulting and ITB Consulting also effectively used lessons learned from reuse cycles to evolve their reusable assets.

> We evolve the components over many life cycles and many iterations of business phases over a period of time. The components have evolved from an architectural standpoint as well as a functional standpoint. Therefore they are continuously updated, I guess, or more up to date when they're done as opposed to being outdated when they are completed.
>
> *Asset Creator, ITA Consulting*

Little discussion surfaced at Global Telecom and Oil & Gas related to organizational learning. This absence of systematic attention to learning tended to be reflected in an undisciplined approach to implementing software reuse in both firms.

> Right now I would have to say reuse is probably fairly difficult because there are not a lot of people that really understand what we're talking about when we use those terms. There's a lot of education that has to happen, I think, for people to understand a little bit about why you want them to do reuse.
>
> *Asset Creator, Global Telecom*

Summary

Taken together, the quantitative and qualitative results support the posited research model. As expected, software development groups experience conflict when implementing architectural changes to development processes dictated by the adoption of software reuse. Those companies (ITA Consulting and ITB Consulting) in which software reuse was viewed as a success achieved these benefits by creating a shared vision across all stakeholders that reuse was a sensible strategy for enhancing application development. Asset creators and asset users (along with other key stakeholders such as managers and clients) were sympathetic to the need to compromise transitory short-term benefits (populating an empty component archive and satisfying demanding clients) to achieve more enduring long-terms benefits associated with enhancing software development productivity through component-based development. On the other hand, software developers at companies in which reuse was viewed as a failure could not commit to the goal of developing systems with reuse. They perceived reuse as increasingly difficult to achieve, given their tactical focus on time to market.

Establishment of such a shared vision (Swanson and Ramiller 1997) in the successful firms did not occur without considerable effort. Coordination strategies were developed and implemented, promoting the development and value of the repository of reusable components. Management intervened by monitoring work practices and by rewarding desired behaviors of asset creators and asset users. Formal and informal communication channels were established to facilitate the flow of information and transfer of knowledge between stakeholders. Finally, organizational learning practices were incorporated within the software reuse program to understand the nature and architecture of reusable assets within salient application domains, to refine work processes associated with both asset creation and asset usage, and to incorporate systematically the learning gained through experience into more robust conceptualizations of reuse architectures, processes, and practices.

Limitations of the Research Design

This study has three major limitations. First, the data were limited because interviews with only four software reuse programs were conducted. While attempts were made to ensure that these reuse programs were established formally, provided with sufficient resources, and staffed by asset creators and asset users, the limited sample size along with the contextual differences across these four sites call for caution in generalizing these findings. Second, no clients were interviewed. Our interpretation of reuse program outcomes as well as the nature of the conflicts that arise among stakeholders would have been enriched by exposure to the views of clients of these application development projects. Third, the heavy representation of IT services organizations—especially in light of the fact that successful reuse programs were found in IT services firms—leads to the possible confounding explanation of organization type. Both ITA Consulting and ITB Consulting likely stand to gain much more from a successful reuse program than do Global Telecom and

Oil & Gas. Not only is client solution development the core business of the two IT services firms, but it is possible that well-developed repositories of reusable components might represent strategic assets.

Nonetheless, the firms likely to benefit most from software reuse are also those most likely to attend proactively to the potential goal conflicts that invariably arise in the early stages of a reuse program and to introduce robust organizational learning practices. As our research objective was not to predict which case sites would be successful and which would not, use of IT services firms as case sites has served to increase the potential richness and relevance of the organizational contexts to examine the study's research questions.

Discussion and Conclusions

This study's research goal was to develop an improved understanding of how organizations might successfully implement and derive value from a disruptive innovation targeted at changing the architecture of work processes to increase efficiency and effectiveness. Such change involved the development of a new technology platform through an upstream process, the exploitation of this platform through a downstream process, and the linkage between these upstream and downstream processes. Within the software reuse context, the upstream process involves populating a repository with maximally reusable software assets, the downstream process involves using these reusable assets in developing client solutions, and the linkage between these processes involves interaction among asset creators and asset users to iteratively evolve flexible assets that can then be integrated to fashion software solutions for clients.

Implications for Research

We anticipate that the insights offered by this study will prove useful to scholars interested in studying disruptive IT innovations in general and, in particular, platform-based innovations in which the architecture of work processes is changed. Not only must a new technological system (i.e., an IT-enabled platform) be introduced into and adapted for the target organizational context, but existing work processes and the linkages among them must be reengineered to exploit the functionality of components provided through the platform. The challenge is heightened by the facts that (1) the creation of the technology platform must adapt to the unique characteristics of the downstream process, as this process depends on the functionality provided by the platform; (2) the upstream and downstream processes must adjust dynamically to each other as the innovation is implemented; and (3) maximum progress in implementing the platform will impair the effectiveness of the downstream process, while maximum effectiveness of the downstream process will impair progress in implementing the platform.

Due to the dynamic, interdependent, and conflicted natures of these two innovation efforts, the actors involved in the upstream (platform construction) and downstream (work process execution) processes must be motivated to contribute to system-level goals as well as to inform and be informed by each other. Such inducements will not occur unless management intervenes (Jasperson et al. 2005) by devising appropriate coordination mechanisms (monitoring, rewarding, and communicating) and orchestrating appropriate organizational learning (conceptual and operational) practices.

In particular, the value of organizational learning practices with disruptive IT innovations cannot be underestimated. At the beginning of such innovation initiatives, stakeholders are likely to hold immature understandings regarding the entire initiative and the substance and structure of facilitating coordination mechanisms. Without substantial investment in and continued attention to organizational learning, improvements in these understandings are unlikely to occur or will occur at undesirable (and likely ineffective) rates.

After theorizing about post-adoptive behaviors associated with information technology enabled work systems, Jasperson et al. (2005, p. 545) encourage scholars

> to apply research designs that enable them to discover, identify, and account for salient interventions directed at all of the work system elements associated with the focal IT application. Research studies that fail to account for such interventions will likely observe considerable unexplained variance.

We believe that this study's findings regarding the nature of facilitative coordination mechanisms and organizational learning practices should assist scholars in their efforts to identify and incorporate within research designs these salient management interventions.

Implications for Practice

With regard to practice, three observations that follow from the study's findings seem most important.

Successful disruptive innovations require management to examine the impact of architectural changes to work processes on the stakeholders involved. It is critical for management to understand the overall technical and business goals being sought, to understand how these goals are being interpreted by stakeholders and how they might induce conflicts within the work processes involved, to identify the critical interfaces between these work processes, and to devise effective tactics for resolving work process conflicts.

Successful disruptive IT innovations demand that a shared vision of an initiative's nature and effects be developed by, communicated to, and accepted by all stakeholders. Such an "organizing vision" (Swanson and Ramiller 1997) must make clear the program's short- and long-term benefits, costs, and risks. Most, if not all, stakeholders are likely to incur some short-term pain to reap the long-term benefits. In particular, managers must realize that short-term goal suboptimization is required to achieve long-term goal optimization. Establishing and then managing stakeholders' expectations through the period of short-term pain is critical.

Successful disruptive IT innovations demand, at their core, active attention to organizational learning. By its nature, innovation involves experiencing and learning about new things. With disruptive IT innovations, such learning requirements are significantly enhanced (Christensen 1997). As a consequence, resources and time must be invested in learning activities, and the learning that occurs must be captured, distributed, and, most importantly, fed back into the innovation initiative and its management.

Conclusion

This study has provided an enriched understanding of the organizational assimilation of a disruptive IT innovation. Focusing on changes introduced in the architecture of work processes, we examined goal conflicts likely to be experienced by interdependent peers and concerted managerial tactics to reduce these conflicts. The conceptual frame and findings should prove useful to scholars studying software reuse, the implementation of IT-enabled (technology or business) platforms, and technology-enabled, disruptive organizational innovations.

Acknowledgments

This research was supported by the Center of Management Information Systems at Texas A&M University and the Rawls College of Business at Texas Tech University. We gratefully acknowledge the helpful comments of Ron Weber, the associate editor, and the three anonymous reviewers of earlier drafts of this manuscript. We also thank Jay Conover, Ron Bremer, Shaila Miranda, Eric Walden, Vasuki Basavanahalli, and seminar participants at the American University, Indiana University, University of Oklahoma, and Texas Tech University for their thoughtful feedback.

References

Abernathy, W., and Clark, K. "Innovation: Mapping the Winds of Creative Destruction," *Research Policy* (14:1), 1985, pp. 3-22.

Abraham, C., and Sheeran, P. "Implications of Goal Theories for the Theories of Reasoned Action and Planned Behavior," *Current Psychology: Developmental, Learning, Personality, Social* (22:3), 2003, pp. 264-280.

Andres, H. P., and Zmud, R. W. "A Contingency Approach to Software Project Coordination," *Journal of Management Information Systems* (18:3), 2001-2002, pp. 41-70.

Argote, L. "Group and Organizational Learning Curves: Individual, System and Environmental Components," *British Journal of Social Psychology* (32), 1993, pp. 31-52.

Arrow, K. J. "The Economic Implications of Learning by Doing," *Review Economic Studies* (26), 1962, pp. 166-170.

Austin, J. T., and Vancouver, J. B. "Goal Constructs in Psychology: Structure, Process, and Content," *Psychological Review* (120), 1998, pp. 338-375.

Austin R. D. "The Effects of Time Pressure on Quality in Software Development: An Agency Model," *Information Systems Research* (12:2), 2001, pp. 195-207.

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

Banker, R. D., and Kemerer, C. "Performance Evaluation Metrics for Information Systems Development: A Principal-Agent Model," *Information Systems Research* (3:4), 1992, pp. 379-400.

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. "The Case Research Strategy in Studies of Information Systems," *MIS Quarterly (*11:3), 1987, pp. 369-386.

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

Carver, C. S., and Scheier, M. F. *On the Self Regulation of Behavior*, Cambridge University Press, Cambridge, UK, 1998.

Choudhury, V., and Sabherwal, R. "Portfolios of Control in Outsourced Software Development Projects," *Information Systems Research* (14:3), 2003, pp. 291-314.

Christensen, C. M. *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*, Harvard Business School Press, Boston, 1997.

Crowston, K. "A Coordination Theory Approach to Organizational Process Design," *Organization Science* (8:2), 1997, pp. 157-175.

Davis, F. D., Bagozzi, R. P., and Warshaw, P. R. "User Acceptance of Computer Technology: A Comparison of Two Theoretical Models," *Management Science* (35:6), 1989, pp. 982-1003.

Dusya V., and Crossan, M. "Strategic Leadership and Organizational Learning," *Academy of Management Review* (29:2), 2004, pp. 222-240.

Dubé, L., and Paré, G. "Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations," *MIS Quarterly* (27:4), 2003, pp. 597-635.

Eisenhardt, K. M. "Agency Theory: An Assessment and Review," *Academy of Management Review* (14:1), 1989, pp. 57-74.

Eisenhardt, K. M. "Control: Organizational and Economic Approaches," *Management Science* (31:2), 1985, pp. 134-139.

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

Fichman, R. G., and Kemerer, C. F. "Incentive Compatibility and Systematic Software Reuse," *Journal of Systems & Software* (57:1), 2001, pp. 45-60.

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

Frakes, W., and Isoda, S. "Success Factors of Systematic Reuse," *IEEE Software* (11), 1994, pp. 15-19.

Gomaa H., and Kerschberg, L. "Domain Modeling for Reuse and Evolution," in *Proceedings of the IEEE 7th International Workshop on Computer-Aided Software Engineering*, Toronto, Ontario, Canada, July 9-14, 1995, pp. 162-171.

Griss, M. L., Kozaczynski, W., Wasserman, A. I., Jette, C., and Troy, R. "Object-Oriented Reuse," in *Proceedings of the 3rd International Conference on Software Reuse* (Rio de Janeiro, Brazil), W. Frakes (ed.), ACM Press, New York, November 1994, pp. 209-213.

Hatch, N. W., and Mowery, D. C. "Process Innovation and Learning by Doing in Semiconductor Manufacturing," *Management Science (*44:11), 1998, pp. 1461-1478.

Henderson, R. M., and Clark, K. B. "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," *Administrative Science Quarterly* (35), 1990, pp. 9-30.

Henderson, R. M., and Cockburn, I. "Measuring Competence? Exploring Firm Effects in Pharmaceutical Research," *Strategic Management Journal* (35), 1994, pp. 63-75.

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

Hoskisson, R. E., and Hitt, M. A. "Strategic Control Systems and Relative R&D Investment in Large Multiproduct Firms," *Strategic Management Journal* (9:6), 1988, pp. 605-622.

Hyland, M. E. "Motivational Control Theory: An Integrative Framework," *Journal of Personality and Social Psychology* (55), 1988, pp. 505-518.

Itoh, H. "Incentives to Help in Multi-Agent Situations," *Econometrica* (59:3), 1991, pp. 611-636.

Jaber, K., Nada, N., and Rine, D. "Product Line Stakeholder Viewpoint Approach and Validation Model," in *Proceedings of the 2000 ACM Symposium on Applied Computing–Volume 2* (Como, Italy), ACM Press, New York, 2000, pp. 871-875.

Jasperson, J., Carter, P. E., and Zmud, R. W. "A Comprehensive Conceptualization of the Post-Adoptive Behaviors Associated with IT-Enabled Work Systems," *MIS Quarterly* (29:3), 2005, pp. 525-557.

Jehn, K. A. "A Qualitative Analysis of Conflict Types and Dimensions in Organizational Groups," *Administrative Science Quarterly* (42:3), September 1997, pp. 530-557.

Jehn, K. A., and Mannix, E. A. "The Dynamic Nature of Conflict: A Longitudinal Study of Intragroup Conflict and Group Performance," *Academy of Management Journal* (44:2), 2001, pp. 238-251.

Jensen, M. C., and Meckling, W. H. "Theory of The Firm: Managerial Behavior, Agency Costs and Ownership Structure," *Journal of Financial Economics* (3), 1976, pp. 305-360.

Karoly, P. "Expanding the Conceptual Range of Health Self-Regulation Research: A Commentary," *Psychology and Health* (13), 1998, pp. 741-746.

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

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

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

Krippendorff, K. *Content Analysis: An Introduction to its Methodology,* Sage Publications, Beverly Hills, CA, 1980.

Kwon, T. H., and Zmud, R. W. "Unifying the Fragmented Models of Information Systems Implementation," in *Critical Issues in Information Systems Research*, R. J. Boland and R. A. Hirschheim (eds.), John Wiley, New York, 1987.

Levinthal, D. "A Survey of Agency Models of Organizations," *Journal of Economic Behavior and Organizations* (9:2), 1988, pp. 153-185.

Locke, E. A., Latham, G. P., and Erez, M. "The Determinants of Goal Commitment," *Academy of Management Review* (13:1), 1988, pp. 23-39.

Lyttinen, K., and Rose, G. M. "The Disruptive Nature of Information Technology Innovations: The Case of Internet Computing in Systems Development Organizations," *MIS Quarterly* (27:4), 2003, pp. 557-595.

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

Montoya-Weiss, M. M., Massey, A. P., and Song, M. "Getting it Together: Temporal Coordination and Conflict Management in Global Virtual Teams," *Academy of Management Journal* (44:6), 2001, pp. 1251-1262.

Mukherjee, A. S., Lapre, M. A., and Wassenhove, L. V. "Knowledge Driven Quality Improvement," *Management Science* (44:11), 1998, pp. S35-S49.

Nelson, R., and Winter, S. *An Evolutionary Theory of the Firm*, Harvard University Press, Cambridge, MA, 1982.

Nilakant, V., and Rao, H. "Agency Theory and Uncertainty in Organizations: An Evaluation," *Organization Studies* (15:5), 1994, pp. 649-672.

Orlikowski, W. J. "Case Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development," *MIS Quarterly* (17:3), 1993, pp. 309-340.

Pich, M. T., Loch, C. H., De Meyer, A. "On Uncertainty, Ambiguity, and Complexity in Project Management," *Management Science* (48:8), 2002, pp. 1008-1023.

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

Prieto-Diaz, R. "Status Report: Software Reusability," *IEEE Software* (10:3), 1993, pp. 61-66.

Purvis, R. L., Sambamurthy, V., and Zmud, R. W. "The Assimilation of Knowledge Platforms in Organizations: An Empirical Investigation," *Organization Science* (12:2), 2001, pp. 117-135.

Ravichandran, T. "Software Reusability as Synchronous Innovation: A Test of Four Theoretical Models," *European Journal of Information Systems* (8), 1999, pp. 183-199.

Rine D., and Sonnemann, R. "Investments in Reusable Software: A Study of Software Reuse Investment Success Factors," *The Journal of Systems and Software* (41), 1998, pp. 17-32.

Rothenberger, M. A.., Dooley, K. J., Kulkarni, U. R., and Nada, N. "Strategies for Software Reuse: A Principal Component Analysis of Reuse Practices," *IEEE Transactions on Software Engineering* (29:9), 2003, pp. 825-837.

Rothenberger, M. A., and Hershauer, J. C. "A Software Reuse Measure: Monitoring an Enterprise-Level Model Driven Development Process," *Information & Management* (35:5), 1999, pp. 283-294.

Sharma, R., and Yetton, P. "The Contingent Effects of Management Support and Task Interdependence on Successful Information System Implementation," *MIS Quarterly* (27:4), 2003, pp. 533-555.

Sherif, K., and Menon, N. "Managing Technology and Administration Innovation: Four Case Studies of Software Reuse," *Journal of the Association of Information Systems* (5:7), 2004.

Sherif, K., and Vinze, A. "Barriers to Adoption of Software Reuse: A Qualitative Study," *Information & Management* (41:2), 2003, pp. 159-175.

Sikora, R., and Shaw, M. "A Multi-Agent Framework for the Coordination and Integration of Information Systems," *Management Science* (44:11), 1998, pp. S65-S79.

Sinclair-Desgagné, B. "How to Restore Higher-Powered Incentives in Multitask Agencies," *Journal of Law Economics & Organization* (15:2), 1999, pp. 418-433.

STARS (Software Technology for Adaptable Reliable Systems). *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, February 1996.

Swanson, E. B. "Information Systems Among Organizations," *Management Science* (40:9), 1994, pp. 1069-1088.

Swanson, E. B., and Ramiller, N. C. "The Organizing Vision in Information System Innovation," *Organization Science* (8:5), 1997, pp. 458-474.

Tracz, W. "Software Reuse: Motivations and Inhibitors," in *Proceedings of COMPCON '87* (San Francisco), IEEE Computer Society Press, Los Alamitos, CA, February 1987, pp. 358-363.

Vitharana, P. "Risks and Challenges of Component-Based Software Development," *Communications of the ACM* (46:8), 2003, pp. 67-72.

Vitharana, P., Zahedi, F., and Jain, H. "Knowledge-Based Repository Scheme for Storing and Retrieving Business Components: A Theoretical Design and an Empirical Analysis," *IEEE Transactions on Software Engineering* (29:7), 2003, pp. 649-665.

Von Hippel, E., and Katz, R. "Shifting Innovation to Users via Toolkits," *Management Science* (48:7), 2002, pp. 821-833.

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

Zmud, R. "Diffusion of Modern Software Practices: Influence of Centralization and Formalization," *Management Science* (28:12), 1982, pp. 1421-1431.

Zmud, R. "An Examination of 'Push-Pull' Theory Applied to Process Innovation in Knowledge Work," *Management Science* (30:6), 1984, pp. 727-738.

About the Authors

Karma Sherif is an assistant professor of Management Information Systems at the Rawls College of Business at Texas Tech University. Her research and teaching interests focus on the adoption of new software development processes and their effect on intra-organizational relationships. She received her Ph.D. in Management Information Systems from Texas A&M University in 1998.

Robert W. Zmud is Professor, Michael F. Price Chair in MIS and Director, Division of MIS, Michael F. Price College of Business, University of Oklahoma. His research interests focus on the organizational impacts of information technology and on the management, implementation, and diffusion of information technology. He is a senior editor with *Information Systems Research* and with the *Journal of AIS*, and he currently sits on the editorial boards of *Management Science*, *Academy of Management Review,* and *Information and Organization*. He holds a Ph.D. from the University of Arizona and an M.S. from M.I.T.

Glenn J. Browne is an associate professor and the Wetherbe Professor of Information Technology at the Rawls College of Business Administration at Texas Tech University. He received his Ph.D. from the Carlson School of Management at the University of Minnesota. His research focuses on cognitive and behavioral issues in information requirements determination, systems development, e-business, and managerial decision making. His articles have appeared in *Management Science*, *Organizational Behavior and Human Decision Processes*, *Journal of Management Information Systems*, and other journals.