

# Barriers to adoption of software reuse

## A qualitative study

Karma Sherif<sup>a,\*</sup>, Ajay Vinze<sup>b,1</sup>

<sup>a</sup>*Information Systems and Quantitative Sciences Department, College of Business Administration, Texas Tech University, Lubbock, TX 79409, USA*

<sup>b</sup>*School of Accountancy and Information Management, College of Business, Arizona State University, Box 873606, Tempe, AZ 85287-3606, USA*

Accepted 23 February 2003

### Abstract

With economic pressures to deliver software applications at a faster rate and at lower cost, software reuse is becoming a significant technology for software development. This paper focuses on deriving a descriptive and explanatory theory concerning the individual and organizational barriers associated with the adoption of reuse. A *case-study* research method was used. A series of five cases were selected on the basis of theoretical replication. The findings, which indicate that barriers occur at both the individual and organizational level, suggest that those at the individual level are actually a consequence of the interaction of barriers caused at the organizational level.

© 2003 Elsevier Science B.V. All rights reserved.

**Keywords:** Software reuse; Component development; Barriers to adoption; Qualitative research; Grounded theory methodology; Case studies

### 1. Introduction

Software reuse has generated much interest in both academia and industry. Though a large number of studies have highly touted the technology [25,29,30,54], outcomes of the adoption process have varied widely [2,21]. Software reuse causes changes in the work of software developers as well as in an organization as a whole [52]. While some organizations are capable of adapting to change quickly, others are likely to resist it and stifle the adoption process.

A review of the reuse literature reveals that there is a paucity of studies to explain contradictory experience with software reuse. Researchers have alluded to the importance of non-technical like human nature and personal characteristics with no empirical evidence to explain interactions among reuse, the individuals involved, and the organizational context [1,18,24]. Experience with previous technology implementations suggest that understanding of the nature of relationships that might exist between the technology and the organizational elements, is necessary to be successful in implementing it.

Following the grounded theory approach, this paper focuses on identifying the individual and organizational factors that affect the adoption of software reuse. The research questions are: (1) why do individuals and organizations constrain the

\* Corresponding author. Tel.: +1-806-742-2163; fax: +1-806-742-3193.

E-mail addresses: [ksherif@ba.ttu.edu](mailto:ksherif@ba.ttu.edu) (K. Sherif), [ajay.vinze@asu.edu](mailto:ajay.vinze@asu.edu) (A. Vinze).

<sup>1</sup> Tel.: +1-480-965-6685.

adoption of reuse? and (2) what are the forms of these barriers?

The study examines the individual beliefs of four reuse stakeholders: *the reuse expert or reuse champion, asset creators, asset utilizers, and IT managers*, with regard to the perceived benefits of the technology and resources required for the implementation of reuse.

We have also studied elements of organizational contexts to examine their effects on the adoption of reuse. Studies [8,9,13,44] suggest that resources, strategies, approaches to evaluating performance and tacit assumptions affect adoption.

Five cases were carefully selected from different industries. They were drawn from three fundamentally diverse industries—oil and gas, telecommunication, and software consulting.

## 2. Barriers to reuse adoption

A review of the reuse literature revealed that top management and software developers are the main instigators of barriers to reuse adoption. Technical and organizational barriers come second.

### 2.1. Top management

While reuse promises a quick time to market and a reduction in cost, it requires an investment in both time and resources to deliver these benefits. The cost of building a reuse program is depicted as exorbitant with no significant short-term returns [7,20,33,41]. An extensive list of the costs that organizations would incur before expecting returns cover the technical issues of developing and managing reusable assets as well as the costs of marketing and maintaining internal support for the program; within these items are the expenses of developing a reuse infrastructure that includes tool support, training and educating staff, and restructuring the organization to adapt to changes in software development. Management, however, may not be geared to invest heavily for long-term benefits, when reuse investment is accounted as overhead cost (not directly tied to application development [22]).

The lack of quantitative measures to assess the benefits and costs [48] is another justification for

management aversion to reuse. Conducting an investment analysis is not straightforward [58]. First, the cost of developing a reusable asset depends on the technical competence of the staff and the rigor of the development methodology used [51]. Second, the benefits will depend on the range of products the asset is planned to assist and the stability of the domains. Neither the costs nor benefits are accurately known. In addition, economic models proposed [19,42] are not easy to understand.

### 2.2. Software developers

Resistance to systematic reuse is also caused by some software developers' belief that reuse will inhibit creativity: the "Not Invented Here" syndrome. Reuse requires latitude on the part of the application developer to use others' work. It also requires trust of the original producer, especially in the case of "Black Box" reuse. The developer needs some assurance that the reusable component will perform well.

Another problem is the belief that reuse is an opportunistic "hunter/gatherer" process that depends on the cognitive abilities of developers to locate the right domains and components. Establishing a systematic reuse program requires an implementation strategy that identifies opportunities for reuse, analyzes domains, builds architectural structures, and develops *built for reuse assets*.

From the asset developer's perspective, reuse may pose some threats. Systematic reuse requires the asset builder to document all assumptions, design rationales, and constraints governing the structure or operations of an asset. Such a Total Quality Management approach would have a significant impact on the development process [16] and time to market the assets that may be avoided when the additional effort seems excessive and of little value to the producer.

### 2.3. Technical barriers

Some of the barriers are directly related to the complexity of the reuse technology. These revolve around three factors: the process involved in building reusable assets, the reusable assets themselves, and the support available for utilizing and maintaining the assets.

### 2.3.1. Problems with development methodology

Problems arising are mainly due to incompatibility of reuse activities with the organization's traditional development methodology. Traditional methodologies like the *Waterfall Model* may not be suitable. The top-down approach has been criticized as inappropriate because systematic reuse requires combination of top-down, *problem-oriented* and bottom-up, *reuse-oriented* perspective [50]. Nor does object-oriented (OO) technology automatically guarantee the creation of reusable objects [27,32,53]: it is up to the systems analyst to plan for reuse when conducting OO analysis and design [37].

Several methodologies have emerged for the development of reusable components and their assembly into applications. These include organization domain modeling and model-based software engineering [49]. An integral part of these is domain engineering, which runs as a pre-process to system development [43] and emphasizes the development of process modeling [45]. It allows for the development of a group of products that share a common set of functionality: a *product line*. The components can be assembled to build different applications.

### 2.3.2. Problems attributed to the assets

The design, implementation, and documentation of reusable components poses additional barriers. Components with a narrow scope or those with a broad range may require excessive efforts in adapting them to a new context. The failure of reusable components to accommodate future needs within a domain is a major contributing factor to the limited success of software reuse [35]. The implementation must go beyond a particular customer's perceived needs.

Another problem is in parameterization. Systematic reuse increases the flexibility of reusable components through the use of parameters. The approach requires extensive testing to ensure that they are not in conflict and will not cause a malfunction [6].

Lack of detailed documentation may also limit use. Usage of reusable assets need the user to know: the *concept*, the *content*, and the *context* of a component [57]. The *concept* describes the objectives, the *content* defines how a component achieves its goals, and the *context* defines where the component can be reused.

### 2.3.3. Problems with supporting tools

The major technical barrier pertains to difficulty of locating reusable assets. Software repositories, while not considered a success factor [17] are believed to be responsible for dissatisfaction with reuse. The structure of the repository may make it hard to locate an asset.

## 2.4. Organizational barriers

Among the organizational elements that constrain the adoption of reuse are the structure of the reuse group and its performance evaluation. The lack of an autonomous reuse group with authority to create and evolve reusable components may lead to it being treated as a side task. Brownsword and Clements report that a key factor to the success of a Swedish company was the establishment of a team responsible for the creation and evolution of software product-line architecture. Several practitioners view resistance as a consequence of “weak collaborative praxes” embedded within the hierarchical divisional structure of the firm inhibiting inter-team collaboration [12,28].

The lack of incentives is another flaw [14]. Unless the producer is reimbursed or otherwise encouraged, there is little incentive to invest time and resources for component reuse. It must be noted that reuse incentives are a “short-term fix for a long-term problem” and should complement efforts to resolve conflicts and control risk factors for software developers [26,56].

## 3. Research design and procedures

Case research is particularly suitable for problems where research and theory are at early stages of formulation [3]. Our study was intended to help find how and why organizations constrain the adoption of software reuse.

An important distinction between case research and other empirical methods is that the variables of interest that explain the phenomena are not identified prior to the study. Both the variables and the relationships between them emerge as the data is collected or analyzed.

By following a grounded theory approach, our first step was to create conceptual categories and establish a context. The categories were drawn specifically from

the reuse literature and in particular from studies related to adoption. Interview questions were designed around these categories. Data, in the form of interview transcripts, were classified and initial classifications were compared among the different reuse groups within and across the five cases. This helped define the scope of the theory and also broadened its explanatory and predictive power. We alternated between data collection and analysis to determine whether new categories needed to be developed [11,23,55].

### 3.1. Research design

A multiple case study approach was adopted to gain an understanding of relationships between the adoption of reuse and the elements of an organization. Embedded units within the organization were considered in explaining the existence of barriers to technology where possible [5]. In each case, two levels of analysis were conducted, the first focused on the affected individuals, while the second addressed the organizational context. The beliefs of four were examined in an attempt to identify attributes that constrain reuse [15]. Two constructs, *benefit of the reuse technology* and *ease of implementing reuse* were utilized as composite surrogate measures of individual beliefs [10]. The industry experts suggested that these were appropriate measures of the stakeholders' willingness to adopt reuse. The stakeholders' feelings that reuse is capable of reducing the cost and time for development of new applications and improving the quality of the end product was a "measure" used to assess value. Considering the ease of implementing reuse, we examined the stakeholders' beliefs of the complexity of building reusable assets and assembling them to build future systems.

A second perspective on reuse focused on IT adoption studies by asking: can elements of the organizational context such as strategies, resources, and culture have an impact on reuse adoption [34,40]? We focused on goal setting, strategy formulation, resource provision, and development of a reuse-oriented culture. We also gathered data on the receptivity of the employees to information sharing, collective achievement, and reuse of knowledge.

The questions for the interviews were originally formulated using a start-up list of concepts identified from the literature. These constructs were not used as a

priori explanations of why barriers exist [31]. New concepts emerged during the analysis stage and these were reinforced by additional data collection efforts.

### 3.2. Site selection

Thirty-five different organizations were initially identified and contacted. A preliminary interview was conducted with at least one individual from each of them to narrow down the list and find cases to illustrate concerns related to reuse. Special attention was paid to covering experience related to both success and failure.

The organizations were selected to help understand the central phenomenon of this study: barriers to adoption. The final selection was based on theoretical rather than random sampling. We attempted to maximize diversity in the cases that we selected: the overriding criteria were the importance of software reuse and availability of documented cases. Five settings from four different companies were included in our final choice. Even though the cases were drawn from different industries, the effort was focused on identifying common causes and forms of barriers. The four sites are referred to as Information Technology Services (ITS), Software Consulting Company (SCC), Production and Operation Management (POM) at Oil and Gas Company (OGC), STAR II, and Telecommunication Company (TCC) to protect their identity.

ITS is an IT service company that focuses on client/server and advanced technologies. The company has 13 different offices with a staff of 1000 employees. They specialize in providing solutions to energy, communication, financial services companies, and government agencies. The average annual revenue during the last 3 years was US\$ 150 million. ITS focuses on providing quick solutions to its customers; their philosophy is to anticipate change rather than react to it.

SSC is a leading software consulting organization that delivers systems to its clients. The study focused on their Energy Solution Group, which provides one product, a gas accounting system, to over 12 customers.

OGC is an oil and gas company with a presence worldwide. It is considered among the 10 largest industrial organization with US\$ 9.8 billion in revenue. The effort focused on the US IT organization within OGC; a company with commercial revenues of around US\$ 750 million. Two groups were studied: the

POM and STAR II is a group that develops reusable assets for the exploration and production field. All of the IT customers are internal.

- At POM, IT projects deal with the chemical engineering aspects; the group employs 65 employees working on about 40 development projects a year. The average budget per project runs around US\$ 100,000. The group made several attempts to launch a reuse program but was not successful. However, in the Spring of 1996, the group started a reuse initiative; a committee was formed to assess reuse opportunities within a specific domain.
- STAR II created a shared repository to serve the common needs of several disciplines (exploration and production, reservoir engineering, seismic interpretation, etc.). At the time of the study, customers viewed STAR II as an exemplar of an initiative that successfully developed and institutionalized a repository of software assets.

TCC is the world's premier communications and information services company, serving more than 90 million business and government customers. The company has annual revenues of more than US\$ 52 billion and more than 130,000 employees. The case focused on the Customer Billing Services (CBS) unit. CBS provides applications for residential customers. The CBS budget for production and development is US\$ 150 million. It delivers hundreds of projects a year with an average budget size that ranges from a few thousand of dollars to US\$ 6 million for major systems with a time to market of less than 3 months.

### 3.3. Data collection

The data collection activity primarily utilized structured interviews. A set of open-ended questions (see Table 1) was posed to each participant at the start of the interview. The aim was to allow participants to express beliefs freely based on their personal experience.

Table 1  
Start-up list of questions for the interviews

Unit of analysis	Question	Theoretical concepts
Individual beliefs	What does software reuse mean to you?	Meaning of reuse
	From your own perspective, what are the benefits of software reuse that you have experienced over the years?	Benefits of reuse
	Do you believe that the technology is easy to implement?	Ease of implementing reuse
	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?	
	What are the main barriers that exist within the organization to software reuse?	Barriers to reuse
	What are the problems that are constraining reuse?	
Organizational readiness	Where do you think reuse fit within the IS goals?	Strategic importance of reuse
	Why do you think your organization is interested in reuse?	Policies
	Are there any policies or strategies that govern reuse?	
	Are you aware of any policies or strategies that unintentionally constrain reuse?	
	What resources are allocated to a reuse program?	Resources
	How is your performance evaluated?	Performance evaluation
	Do you think the organizational culture values individualism more or collectivism?	Culture
	Are people in the organization willing to share their experience freely with others?	Rewards
	Are organizational members willing for the organization to capture their knowledge?	
	Are there formal organizational incentives for reuse stakeholders to adopt reuse?	
	Does the organization reward members for sharing knowledge?	

Table 2  
Type and number of interviews conducted at each site

Position	ITS	SCC	POM at OGC	STAR II at OGC	TCC
Reuse expert	1	1	3	1	2
Asset creators	2	2	2	1	2
Asset utilizers	1	2	4	1	2
Project managers	1	0	<sup>a</sup>	1	1
Managers	<sup>a</sup>	<sup>a</sup>	1	<sup>a</sup>	1
Total	5	5	10	4	8

<sup>a</sup> The reuse expert also acted as a senior manager and provided input based on his role in top management.

After the initial round of interviews, a new set of questions was added to reflect additional concepts that emerged. Follow-up interviews were conducted to collect data on emerging concepts that were not considered in original interviews. This approach is considered legitimate in grounded theory methodology and has been successfully adopted in IT research [39].

A total of 32 interviews were conducted: 28 were taped and transcribed. With the permission of the participants, extensive notes were taken. The distribution of interviews among the five organizations is shown in Table 2. Our goal was to slice vertically through each organization and obtain data from multiple levels and perspectives.

### 3.4. Data analysis

Data analysis was iterative. In fact, we alternated between data collection, coding, and data analysis to uncover new sources of data [59]. The data analysis started with the transcription of each interview and the comments recorded during the course of the interview [36]. Though the data collection started with a set of predefined codes, every interview was carefully read to extract new codes. The QSR NUD\*IST software<sup>2</sup> was used to categorize sentences. We followed an *open coding* approach, as recommended in grounded theory methodology, to build evidence for the emerging codes.

Earlier cases were reconsidered in the light of the kind of relationships that existed among stakeholders;

the social and organizational role of the reuse champion; the effect of clients' behavior on the adoption of reuse; plus the organization's business philosophy regarding short versus long-term goals and ways of attaining them. Concepts were grouped into categories, a process known as *selective coding*. Occasionally, new categories were added and others were regrouped when transcribed interviews were analyzed. After comparison of all interviews, a saturated list of categories was produced. Data collected for each category was compared within and across sites: the main intent was to compare and contrast the causes and forms of barriers to reuse adoption.

At the second level of analysis, a connection between the categories was developed through a *Paradigm Model*. This involved identifying the *causal conditions* for the *phenomena*, the *context* surrounding a phenomenon, the structural conditions that affect the relationship between phenomena and the causal conditions, the strategies taken to respond to the phenomena, and the *consequences* of the interaction of the meta-constructs.

From this, it became apparent that the causal condition for the existence of barriers to reuse adoption was management philosophy due to short-term goals. This caused shortages of resources for reusable assets. Several contextual and structural factors played a role in dampening the negative effect, one was the existence of a visionary reuse expert who led the effort to procure the necessary endorsements for the success of the reuse program. The stakeholders' attitudes may have been a consequence of actions taken by both the reuse expert and management.

## 4. Findings

The comparative analysis of the data collected from all five sites revealed a consistent set of concepts. The data suggests that barriers to reuse occurred throughout the spectrum of the life cycle.

### 4.1. Management philosophy

The literature strongly suggests that reuse initiatives will result in value only when organizations intervene to modify their business philosophy [4]. In all five cases, there was evidence that the primary barrier to the creation, use, and management of reusable assets is

<sup>2</sup> N4 (non-numerical unstructured data indexing searching and theorizing) qualitative data analysis program, version 4, QSR International Pty Ltd., Melbourne, Australia, 1998.



management's reluctance to allocate resources to build the infrastructure and its goal on rapidly delivering solutions to market. In all five organizations, reuse stakeholders accused their top management of a tactical focus on time to market without due considerations to long-term benefits gained from the adoption of systematic reuse. We thus state the following proposition.

**Proposition 1** (Business philosophy). *A business philosophy focusing on the short-term goals of delivering projects on time is the main cause of barriers to reuse adoption.*

At all five sites, the focus on short-term goals diverted attention and resources away from the systematic development of reusable components. The only exception was the initial stages of the STAR II project in the oil and gas company; it was launched for the sole purpose of establishing a reusable architecture for the sub-surfacing domain and populating the reuse repository to benefit the development effort of that application groups' domain. Towards the end of the study, STAR II was losing support and was then

converted to a cost and profit center, to be solely financed by its clients. Summary of the findings are given in Table 3.

#### 4.2. Reuse champions

Despite the limitations imposed by the business philosophy, three of the cases studied managed to continue their reuse projects beyond the initiation stage. Their success was heavily dependent on the availability of reuse champions [47], but data collected from the study suggested that the zeal of the whole reuse group made more difference than that of the champion. A reuse expert may ignite the initial spark but reuse efforts are not sustainable without a system of "partnerships" among creators, utilizers, and supporters who have the authority to set priorities and make decisions.

At SCC, STAR II, and ITS the champions were devoted to reuse and were highly respected by their clients, mainly application groups. In all three cases, they took responsibility for mentoring the developers and integrating reusable assets. They adopted strategies to build up the technical skills for reuse.

Table 3  
Support for the business philosophy proposition

Site	Degree of support to the proposition	Excerpts of data supporting the proposition
SCC	High	It is more likely the setup of SCC IS based on a project base rather than on a product in general that we have built. SCC's general approach is to go to a client and come to a specific solution for that client, and implement it
ITS	High	Many barriers to reuse are tactical blinders that focus on the delivery of short-term, immediate results like, "we're going to get this project out quick and get the check," we lose our strategic vision, that we want 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. Those are the things that really screw up reuse, when people just look at the immediate payoff and not at what it can mean for them in the long run
TCC	High	It is the attitude that our organization has, the willingness to sacrifice everything for the time to market that is essentially a constraint. And while it is 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"
POM	High	The way that we manage projects hinders reuse in that we are trying to shorten the delivery time, and forcing the developers to deliver a code as quickly as possible without allowing them time to check with fellow colleagues that may have developed some software that is similar or doing some investigative work on how they can develop a piece of software that can be reused by the next project. So, the hindrances are our main emphasis for delivering software on time for a specific project rather than trying to look at the whole spectrum of projects that are out there
STAR II	High	Recently, we have been a profit and loss center where we actually responsible for turning a profit, which means we care very much about the bottom-line. That puts more of a shortsighted look on reuse. It becomes more difficult now to convince management to take a long-term approach with building a reuse infrastructure of this magnitude

In addition, the availability of talented asset creators played an important role. Those who understood the concepts of reuse and knew how to implement it were inherently inclined to develop reusable assets. Excellent communication skills were needed to help utilizers comprehend the design rationale encapsulated within the assets and to seek feedback concerning

the performance of the assets and potential modifications to improve client productivity. We thus make the following propositions (see the summary in Table 4).

**Proposition 2** (Reuse champions). *A visionary reuse champion can moderate the impact of business philosophy on the existence of barriers.*

Table 4  
Support for the reuse champion propositions

Site	Degree of support to the proposition	Excerpts of data supporting the proposition
SCC	High	<p>Steve keeps us focused on reuse at all the stages of development. He has developed the reuse architecture for us to use during the development of new applications. I mean there was a focus from the very beginning on achieving reuse. It was not an afterthought, which I think really made a big difference because everybody recognized it. Within this domain we could be achieving very high levels of reuse (Proposition 2)</p> <p>We got good people who are doing the architecture and are building reuse into the system (Proposition 3)</p> <p>I have found that people are very willing and more than open to share and help the new comers learn how to reuse and how to build for reuse. They are their comrades (Proposition 4)</p>
ITS	High	<p>I am definitely a proponent of reuse-based development. Within our group, reuse is kind of a shared vision. We have gotten projects and we proved to be successful. We are willing to take on the learning and the mentoring role (Proposition 2)</p> <p>Our developers are grown up in this reuse culture and they are taught from day one the benefits of reuse. They are very amenable to accepting those types of development standards and very quickly start promoting those concepts themselves. We have people who are just dedicated to the reuse initiative even with standards that are not completely defined (Proposition 3)</p> <p>I also appoint mentors on various topics, whose primary responsibility is to own that component and ensure that everybody else is aware of it, how it is used, and how it should be used. And that kind of helps to get people to seek advice or seek experiences from those various mentors in order to get a piece of work done rather than sitting there and inventing it or figuring it out themselves and wasting valuable time (Proposition 4)</p>
TCC	Low	<p>Individual projects are so tightly tied to meeting schedules and deadlines that it is difficult to get them to look beyond their own project's needs, and it is difficult to get them to put in any effort whatsoever to develop a component that might be reusable outside their group (Proposition 2)</p> <p>It is very difficult to get people to adopt it. They, they lip service through it. They will say, "Yes, we realize that it's important," but when it comes right down to dedicating two, three, four people to run a reuse group, that is when it gets difficult. I think the implementation is very difficult (Proposition 3)</p> <p>They have given us tools to do that, they have given us several large areas to share information. They are real big on that (Proposition 4)</p>
POM	Low	<p>There is not really a global focus on what reuse between projects can be because we are under such cost pressures, that we do not put in the necessary effort to do reuse (Proposition 2)</p> <p>It is harder to build reusable components than I to build something specifically for one use (Proposition 3)</p> <p>You have to communicate and facilitate information sharing to let people know about components and help them apply these reusable components. That is one of the challenges we have to do to make reuse happen (Proposition 4)</p>
STAR II	High	<p>We had the foresight to see that reuse will pay in future, that we are not going to see instant results and that is very difficult, because sometimes, especially management in an oil company, they are not all computer background (Proposition 2)</p> <p>A critical success factor was having the right staff skills in place to execute, having people with the right level of experience with object-oriented technology, the domain knowledge and the distributed object knowledge to actually make it successful (Proposition 3)</p> <p>We have a good information sharing system in place. Besides that we try to include as many people as possible in our design reviews (Proposition 4)</p>



**Proposition 3** (Reuse champions). *Asset creators with experience in the domain of focus and necessary skills to analyze and design the domain can moderate the impact of business philosophy on the existence of barriers.*

**Proposition 4** (Reuse champions). *Active communication between asset creators and asset utilizers can moderate the impact of business philosophy on the existence of barriers.*

The reuse experts at TCC and POM at OGC believed that reuse will only work through a top-down approach, while management felt that investment in reuse would be risky without a success story. At ITS and SCC, the experts did not acquire extensive investment for their reuse initiatives. As both firms operate within a highly competitive industry, the inability to leverage internal expertise would have been detrimental to success. Both cases depended heavily on their success stories to break management's skepticism and are trying to make sure that at least the reuse group has adequate discipline, tools, and processes in place to support reuse.

#### 4.3. Barriers to reuse adoption

Barriers manifested themselves during the three stages of the reuse life cycle. The main ones during the early stages of *asset creation* were lack of reuse policies to govern the creation of reusable assets, lack of funding, immature technology, and lack of reuse-oriented education and training. In the *asset utilization* phase, reuse stakeholders complained about difficulty in accessing an asset, limited marketing of assets available in the repositories, and lack of support for the integration of assets during application development. Though stakeholders at all five sites acknowledged an ongoing effort to manage assets, they reported the same barriers that emerged during the creation of assets for the limited role of *asset management*.

##### 4.3.1. Asset creation

The creation of reusable assets is basically a question of finding the commonalties that exist between systems in a specific domain. This depends on the availability of domain knowledge (detailed

information about the entities and relationships between them) and the technical experience of staff to analyze and design solutions for a window of applications rather than just one. Among the barriers in this stage are lack of reuse policies on the creation of reusable assets, lack of funding, immature technology, and lack of reuse-oriented education and training. A summary of our findings is given in [Table 5](#).

**4.3.1.1. Reuse policies.** At SCC, STAR II, and ITS, the reuse experts established a set of standards that helped in the development of reusable assets and enabled the developers to pin a system-specific requirement at the right place in the overall domain architecture. The availability of reusable components and their tight coupling to the reuse architecture made it possible to enter a client site with a portfolio of more than 50% of the clients' functionality needs already built. The reuse architecture enabled these groups to create additional vertical content within the industry segments on which they were focusing. With a system development methodology that incorporated reuse throughout the development cycle, embedding links to a pre-built library of components and kernels of accumulated best practices and procedures, reuse is enforced with every new implementation. In all three cases, these policies were not mandated by administration but were used because of the dedication of the reuse group to the vision.

At TCC and POM at OGC, no reuse policies were defined. This had an adverse effect. There were no standards to govern the creation of assets adaptable to different contexts, leaving the developers in the position of having to take advantage of existing components, altering them to fit within the new applications. Findings at the five sites lead us to propose the following proposition.

**Proposition 5** (Asset creation—reuse policies). *The lack of well-defined reuse policies that cover the technical and managerial aspects of reuse is a barrier to adoption.*

**4.3.1.2. Funding.** Another barrier was the lack of funding to allocate staff to the development of reusable assets. STAR II was an exception; resources were allocated for the sole purpose of populating the reuse

Table 5

Support for barriers to asset creation propositions

Site	Degree of support to the proposition	Excerpts of data supporting the propositions
SCC	Low	<p>We have are guidelines for developing reusable software and standards that encourage people to develop in ways that make components reusable (Proposition 5)</p> <p>We get clients to buy a system that we have not built yet. Each client pays a certain amount on a yearly basis for what they call base version enhancements, as we evolve and create new reusable components we roll out those changes to everyone (Proposition 6)</p> <p>We have a pretty rigorous training program that people go through to teach them the concept of reuse (Proposition 7)</p> <p>Technology is kind of a double edged sword, it lets you do almost anything, which is good, and it is also bad. It does not guarantee that assets developed will be reusable (Proposition 8)</p>
ITS	Low	<p>We created those policies for building a reusable component and, common architectures (Proposition 5)</p> <p>We do have a pretty good budget for internally developing shared components across teams that is partially funded by management and partially the clients (Proposition 6)</p> <p>We take new people and churn them through these 4–6-week classes as they come on into ITS and then either have them available as competent developers for the component and, or put them out into the field with an understanding of what the component center is providing for them and they should start leveraging that immediately (Proposition 7)</p> <p>I does not think that the technology is the biggest issue, but it certainly makes things easier if everything that you develop in an organization follows the same basic architecture, and that architecture makes it easy to plug components into the program (Proposition 8)</p>
TCC	High	<p>We do not have any formal policies available to facilitate reuse, we do not have a lot of discipline around that right now (Proposition 5)</p> <p>Reuse is a multi-tiered activity; some is built and funded by the user dollars, another kind of reuse is a “capability reuse.” This may be infrastructure investment, which includes architectural constructs, components, processes, data repositories, etc. Typically, this is funded by the CIO, having carved out some money for such initiatives (Proposition 6)</p> <p>There is no training except on the job that I have ever seen that supports it. If anything I would say that universities probably go the other way on supporting reuse (Proposition 7)</p> <p>Anytime that we have used OO in our major applications, it becomes a nightmare over time because of the inheritance, the inheritance which starts out as being a real saver as far as testing goes, ends up being a land mine as the system matures and changes (Proposition 8)</p>
POM	High	<p>There is probably not any formal guidelines that covers software development for reuse (Proposition 5)</p> <p>The problem is, the way we work around here is you do work when you get a charge number. So when a client requests an application or something out in the field, then you get a charge number. But just to build a component, you are not going to ever get a charge number for it (Proposition 6)</p> <p>We cannot afford the time to train people to go to that repository when they are needing new components, and to identify needs for new components, and to spend time documenting components that are within their application that they think might have value somewhere else (Proposition 7)</p> <p>A lot of the tools that [they] have right now more project oriented. [They] do not really have a shared component library that is shared across the entire organization (Proposition 8)</p>
STAR II	Low	<p>We also put in place some policies around what we called the requirements classification of essentially pinning a system requirement to the right place in our overall reuse infrastructure (Proposition 5)</p> <p>In terms of actual tools that we used, we did not have any good tools at the time (Proposition 6)</p> <p>Several years ago, we did have our own budget to build the reuse infrastructure. Now we are completely funded by the applications group (Proposition 7)</p> <p>We have extensive training. They send out bulletins of the time where you can check on-line what training courses are provided (Proposition 8)</p>

library, but when STAR II was converted from a cost center to a cost and profit center, pressure was put on the group to generate profits every year. Thus, it is entirely funded by the application groups in return for assets created for them, an arrangement that usually creates barriers because clients are not driven by the concept of reuse, being interested only in getting a system that fits their requirement specifications on time and within budget.

At SCC and ITS, the reuse experts signed contracts committing their customers to finance the development of reusable assets. The reuse group promoted system development to Energy Solution Group customers for support of creation of reusable assets. The clients were required to pay for the base version plus extra annual funds to obtain base version enhancements. This support fee finances evolution of the asset database. The advantage to the Energy Solution Group is that it owns the solutions it develops for customers. The clients realize the benefits of sharing a version that is constantly improving: “it’s kind of like a pool of companies that are all pitching in to enhance one product.” These findings lead us to propose the following proposition.

**Proposition 6** (Asset creation—funding). *The lack of a source of funding for the development and promotion of reuse is a barrier to reuse adoption.*

**4.3.1.3. Education and training.** At ITS, SCC, and STAR II there was specialized training for reuse as well as a rigorous mentoring program to train developers on the concepts of reuse. The reuse expert at ITS even described mentoring as “more important than training” because it helps new developers see ways of reusing existing assets or building for reuse more vividly, especially when staff members are very young and receptive to new techniques.

At TCC and POM, the reuse experts complained of a lack of skills, but no effort was made to educate the developers on reuse. At TCC, the reuse expert said:

There’s a lot of education that has to happen. It just sounds like reuse is the latest buzzword. Education will get developers to understand what it really means. Otherwise the goals can become empty words if you’re not truly backing it up with the time and the education and the processes around it.

We thus state the following proposition.

**Proposition 7** (Asset creation—education and training). *The lack of an educational and training program or a mentoring one that is targeted to reuse is a barrier to reuse adoption.*

**4.3.1.4. Technology.** Technical immaturity affects management’s willingness to invest in it. Developers at all five sites indicated their belief that reuse technology is not yet sufficiently well defined to provide an adequate foundation for a detailed asset development methodology. They also believed that the instability of the technology is one of management’s primary rationales for dropping support of large-scale reuse programs. Two strategies are viewed as helping to mitigate these effects: developing a domain layered architecture and following an evolutionary approach to the development of reusable assets.

All the stakeholders at SCC, STAR II, and ITS believed that a layered architecture that is sufficiently transportable to be applied to different situations is a critical success factor. Asset utilizers at SCC felt that their domain-specific architecture was helpful in developing applications because it guided the integration of reusable assets into a cohesive subsystem. At STAR II, the group developed a domain-specific architecture and engineered “the entire infrastructure from the ground up . . . to provide a clear direction for applications.”

One of the technologies repeatedly mentioned as an enabler for reuse is object-oriented technology. Developers interviewed at STAR II, ITS, and SCC said they believed object-oriented technology to have all the constructs necessary for creating and utilizing reusable assets. Experienced developers, however, commented that:

I think that OO has helped software reuse tremendously within a project, but I don’t think that OO itself has done anything for software reuse across projects. I think that components have done a lot more towards that advice.

One interesting observation unique to CBS at TCC is a belief that OO is barrier to reuse. One project manager expresses her concerns as:

Anytime that we have used OO in our major applications, it becomes a nightmare over time

because the inheritance, which starts out as being a real saver as far as testing goes, ends up being a land mine as the system matures and changes.

At TCC and POM, technology was believed to be a delimiting factor for the adoption of reuse. Neither cases had a consistent methodology to follow for software development. In addition, methodologies did not support the creation of reusable assets or actively consider the reuse of existing ones. We thus state the following proposition.

**Proposition 8** (Asset creation—technology). *The lack of a technology that supports reuse is a barrier to reuse adoption.*

#### 4.3.2. Asset utilization

Asset utilizers complained of lack of information on the availability of reusable assets, difficulty in locating them in the library, and lack of support in integrating them within applications. A summary of the data that emerged appears in Table 6. We thus state the following propositions.

Table 6  
Support for barriers to asset utilization propositions

Site	Degree of support to the proposition	Excerpts of data supporting the propositions
SCC	Low	<p>Our central repository is located here and at each one of our client site we make sure that they are set up to dial up to have access to the repository. However, we do not have any capability that you can query, “I want something that does this, this, and this,” and it will come back and tell you where that is (Proposition 9)</p> <p>We also have a monthly meeting and we talked about things that are developed to make developers more aware of what things are out there (Proposition 10)</p> <p>Whenever we develop a new component all of the developers get a description. Whoever develops a component would then, as the first step build a use for it, and hence, everyone would have an example of what it does and how it does it (Proposition 11)</p>
ITS	Low	<p>We are currently developing some catalogues and pattern repositories to disseminate the knowledge for the components that we are building within the company so that project managers know what is available (Proposition 9)</p> <p>We need to improve our system of advertising components to let developers know what is available (Proposition 10)</p> <p>We have a very large project communication infrastructure in LOTUS notes that we bring to every client or every customer that we work with (Proposition 11)</p>
TCC	High	<p>There is no area that we can look in to search and see if we can reuse a component (Proposition 9)</p> <p>Unfortunately what we lack is we do not have any way to effectively advertise these resources (Proposition 10)</p> <p>One of the real problems is trying to find documentation on what kind of reusable components we have and who supports them (Proposition 11)</p>
POM	High	<p>There is not really a central reuse library at this point. At the current state it is very difficult to locate reusable assets (Proposition 9)</p> <p>We really rely on word of mouth to find about reusable components (Proposition 10)</p> <p>People did not really know who to turn to get support for reusing an object (Proposition 11)</p>
STAR II	Low	<p>We have our libraries partitioned up by functionality so that all of our data objects are in one library, all of our graphic objects are in a different library, all of our classes that involve reading from specific formats, like LANDMARK or CHARISMA, are broken up into their own separate libraries (Proposition 9)</p> <p>We gave a number of workshops that described the platform and what it contained as well as a number of presentations at different times (Proposition 10)</p> <p>We supply lots of example programs that show how to use the infrastructure (Proposition 11)</p>

**Proposition 9** (Asset utilization—accessibility). *The lack of a well-organized and indexed software repository with appropriate search engines is a barrier to reuse adoption.*

**Proposition 10** (Asset utilization—promotion). *The lack of a promotional program that promotes existing assets and their use is a barrier to reuse adoption.*

**Proposition 11** (Asset utilization—support). *The lack of support to assist asset utilizers access, understand, and integrate reusable assets is a barrier to reuse adoption.*

The problem at several sites was that developers were unaware of assets at an organizational level and relied heavily on word of mouth to find them. They asserted that the lack of software repositories and the lack of ways to promoting reusable assets constituted significant barriers. At CBS-TCC and POM-OGC the main problem was the lack of a central repository with a good search mechanism and catalogue system. Developers developed an attitude that: “it was more trouble to find and use than to build on your own.”

At SCC, asset utilizers complained they were not aware of reusable components availability despite a central on-line index. They commented that: “it is the asset creators’ responsibility to inform us of what is

there,” and provide tools to make it easily accessible. They suggested a knowledge-based repository.

At STAR II, the accessibility of the assets was not felt to be a problem. Developers said they believed that assets could be easily found within the repository because the library was categorized under domains; also web pages that search components in the library have been published. The group also provided a use scenario to help asset utilizers understand what each component did and how it did it.

#### 4.3.3. Asset management

Management of reusable assets is an important stage in the adoption of reuse because it helps the group maintain the client’s interest. Ideally, the reuse group would be engaged in an ongoing effort to improve the quality of the assets, based on asset utilizers’ evaluation of the asset performance. Such feedback to creators regarding additions or modifications needs to be incorporated to satisfy customer needs. As the assets go through new reuse cycles, their quality is enhanced in terms of maintainability and adaptability. According to the evidence, we propose the following proposition.

**Proposition 12** (Asset management). *Failure to periodically update assets to improve their quality is a barrier to reuse adoption.*

Table 7  
Support for barriers to the asset management proposition

Site	Degree of support to the proposition	Excerpts of data supporting the propositions
SCC	Low	As the system matures, we have much better understanding of what it is we really want 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 we abstract certain business rules, even the technical architecture, in such a way that we can more easily adapt to change (Proposition 12)
ITS	Low	I am definitely a proponent of an evolutionary approach. One may start or add to an existing component, but that component has evolved over many life cycles or many iterations of business phases or functionality over a period of time (Proposition 12)
TCC	Low	Our biggest problem is that we do not do enough of the up front work. We are not doing a lot of the modeling correctly. We are not doing enough of it to build the component and know that it is going to fit within the architecture going forward (Proposition 12)
POM	Low	We do 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 he next project that comes along, we could extend that object. So not only is the library evolving, but the objects themselves are evolving from project to project (Proposition 12)
STAR II	Low	We are now trying to fill in the gaps, and complete our tool kit (Proposition 12)

A summary of the findings appears in Table 7. At STAR II, the group had been engaged in an ongoing effort to evolve the assets to maintain the interest and support of the application groups; the assets had been through two evolutionary stages during their 8-year lifeline. The group always attempted to take advantage of proven new technologies to increase performance. The emphasis of the group is on filling in gaps and completing the tool kit. As one manager explained:

The overall strategy is to put the extra effort into re-engineering, to fill in the framework properly. On occasion, if the application team needs a specific component early, they will build it in a less generic fashion, and then replace it later with a STAR II components when it becomes available.

It is not only commitment to reuse that drives this group toward evolution of assets but also a firm belief that reuse is more than just building an infrastructure and creating reusable assets while continually evolving that structure to higher standards.

The issue of evolution of reusable assets at POM was purely opportunistic and was accomplished through project cycles during which specific assets were modified to meet the special needs of the application team. Accordingly, it became difficult for developers to integrate a component into new projects without serious design.

Barriers to the evolution of assets were the same as those constraining their creation. In addition, one problem is directly attributed to members of the reuse group: the lack of formal reuse metrics. There has been no attempt at any of the sites to identify the cost and benefits of reuse, making it hard for management to buy-in. The general belief regarding metrics was:

... metrics are tough, though, because they are basically made up .... It's easy to throw some numbers on a new graph and show it to management. If you really know how the metrics is collected then you really know how active they are. They're not particularly useful.

(Asset creator at STAR II)

All the reuse groups measured success in terms of time to market reuse-oriented applications. This is consistent with Frakes and Fox findings that the collection of metrics is not commonly carried out

by reuse programs and is not considered a significant factor in their success.

Asset utilizers required quality assurance and knowledge of the testing techniques, as an alternative to metrics, before fully committing to reusing assets. Feedback on the performance of the asset in projects and satisfaction of application developers with them were crucial for gaining the developers trust in the library. Asset utilizers would also like to see a description of the design pattern encapsulated within the assets to help them assess the applicability of a specific asset to an application.

The reuse experts and managers at SCC and ITS believed that metrics are mostly required for calculating monetary rewards. Rewards, they believed, is not a requirement, especially when developers are so committed to reuse that it becomes a shared vision and, eventually, "the only mindset that the developers have." In fact, managers argued that monetary rewards would add an unnecessary layer of administration to keep track of who develops what and who reuses what. Managers believed that projects should be the unit of analysis: "what's rewarded is the end product not the components of it" as one manager said.

## 5. Validating the framework

Our findings were presented to a panel of four experts on software reuse. Each is a practicing professional in the software industry and has provided consulting services to several organizations that adopted reuse. Three of the experts have written an industry-accepted technical report on software reuse.

The panel provided four suggestions for further enhancing our model. While generally agreeing with the findings, they indicated a need to produce a more complete understanding of the makeup of the "business philosophy" factor. They felt that business philosophy was not the main cause of reuse barriers. Expanding further, they indicated that the real barrier was that "reflection on past experience is painful." Developers typically do not want to take the time to reflect on and learn from past experience, even for an hour. While this is an interesting assertion, our data do not support it. One of the experts commented on the lack of a structured methodology. It was his belief that reuse had to be adopted in a bottom-up approach.



A second suggestion was that “organizations must believe in slowing down to speed up.” When an organizational culture starts to recognize the importance of learning and reflecting on past experience, reuse will start to make sense. Finally, the panel said that an important contributor to resistance is the lack of an accounting system that makes explicit the cost and value of reuse. The existence of a reuse champion who encourages developers to do small pilot studies and produce metrics that convince managers was also believed to be a critical factor.

The second stage of validation involved analyzing a number of information technology research projects on successful adoption of new technology. Internal factors include individual adjustments, organizational receptivity, and technology characteristics. Additionally, Nelson [38] identified satisfaction, involvement, organizational commitment, and performance as elements affecting individual adjustment. Factors related to organizational receptivity included organizational structure, organizational politics, organizational culture, management process, management support, the nature of corporate systems, the quality of human resources, and availability of resources [46]. Issues such as market conditions, competition, regulations, and relationship with government have also been reported as concerns related to the adoption of IT.

A number of factors that emerged from our model previously have been shown to be indicators for outcomes of an IT adoption process. These include business philosophy, technology, staff, funding, and education and training. The added benefit of our framework is that it illustrates relationships between factors rather than depicting them as discrete causes for the success or failure of adoption.

## 6. Summary and conclusion

The data collected from five sites suggested that stakeholders’ beliefs regarding reuse are contingent on organizational actions with respect to the resources required to develop and market reuse. Such beliefs determine the level of commitment of the stakeholders, which ultimately affects the culture.

Our research focused on the development of a grounded theory of barriers to adoption. It sought to build a theoretical framework, seeking data from

different sites to add complexity and generalizability to the theory. This effort makes an important contribution to the study of reuse adoption, and particularly to barriers to adoption. The model indicates that barriers are mainly caused by business philosophy. The persistence and commitment of the reuse champions within the organization moderate this causal relationship. The forms of barriers are the outcomes of organizational actions with respect to the resources required. Consequences of the barriers are the individual beliefs regarding benefit and ease of implementing reuse. A summary of the findings was presented in the form of 12 propositions.

## References

- [1] U. Apte, C.S. Sankar, M. Thakur, J.E. Turner, Reusability-based strategy for development of information systems: implementation experience of a bank, *MIS Quarterly* 14 (4), 1990, pp. 421–433.
- [2] P. Basset, *Framing Software Reuse: Lessons from the Real World*, Yourdon Press Computing Series, Upper Saddle River, NJ, 1997.
- [3] I. Benbasat, D.K. Goldstein, M. Mead, The case research strategy in studies of information systems, *MIS Quarterly* 11 (3), 1987, pp. 369–386.
- [4] T.J. Biggerstaff, A.J. Perlis, *Software Reusability: Applications and Experience*, vol. II, ACM Press, Reading, MA, 1989.
- [5] L.J. Bourgeois III, K.M. Eisenhardt, Strategic decision processes in high velocity environments: four cases in the microcomputer industry, *Management Science* 34 (7), 1988, pp. 816–835.
- [6] L. Brownsword, P. Clements, A Case Study in Successful Product Line Development, CMU/SEI-96-TR-016, Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, PA, 1996.
- [7] J. Cardow, Issues on software reuse, in: *Proceedings of the IEEE National Aerospace and Electronics Conference*, Dayton, OH, 22–26 May 1989, pp. 564–570.
- [8] P.K. Chau, K.Y. Tam, Factors affecting the adoption of open systems: an exploratory study, *MIS Quarterly* 21 (1), 1997, pp. 1–24.
- [9] R. Cooper, R. Zmud, Information technology implementation: a technological diffusion approach, *Management Science* 6 (2), 1990, pp. 156–172.
- [10] F. Davis, R. Bagouzi, P. Warshaw, User acceptance of computer technology: a comparison of two theoretical models, *Management Science* 35 (8), 1989, pp. 982–1003.
- [11] K.M. Eisenhardt, Building theories from case study research, *Academy of Management Review* 14 (4), 1989, pp. 532–550.
- [12] D. Fafchamps, Organizational factors and reuse, *IEEE Software* 11, 1994, pp. 31–41.

- [13] R.G. Fichman, C.F. Kemerer, Adoption of software engineering process innovations: the case of object orientation, *Sloan Management Review* 34 (2), 1993, pp. 7–22.
- [14] R.G. Fichman, C.F. Kemerer, Incentive compatibility and systematic software reuse, *Journal of Systems and Software* 57 (1), 2001, pp. 45–60.
- [15] M. Fishbein, I. Azjen, *Belief, Attitude, Intention and Behavior*, Addison-Wesley, Reading, MA, 1975.
- [16] L.Y. Fok, W.M. Fok, S.J. Hartman, Exploring the relationship between total quality management and information system development, *Information and Management* 38 (6), 2001, pp. 355–371.
- [17] W. Frakes, C. Fox, Sixteen questions about software reuse, *Communications of the ACM* 38 (6), 1995, pp. 75–115.
- [18] W. Frakes, C. Fox, Systematic software reuse: a paradigm shift, in: *Proceedings of the Third International Conference on Software Reuse*, Rio de Janeiro, Brazil, 1–4 November 1994, pp. 2–3.
- [19] W. Frakes, C. Fox, C. Terry, Reuse level metrics, in: *Proceedings of the Third International Conference on Software Reuse*, Rio de Janeiro, Brazil, 1–4 November 1994, pp. 139–148.
- [20] W. Frakes, C. Fox, S. Isoda, Success factors of systematic reuse, *IEEE Software* 11, 1994, pp. 15–19.
- [21] W. Frakes, C. Fox, W.B. Biggerstaff, R. Prieto-Diaz, K. Matsumura, W. Schaefer, Software reuse: is it delivering? in: *Proceedings of the 13th International Conference on Software Engineering*, Austin, TX, 13–16 May 1991, pp. 52–59.
- [22] M.L. Griss, W. Kozaczynski, A.I. Wasserman, C. Jette, R. Troy, Object-oriented reuse, in: *Proceedings of Third International Conference on Software Reuse*, Los Alamitos, CA, 1994, pp. 209–213.
- [23] B.G. Glaser, A.L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine Publishing Company, New York, NY, 1967.
- [24] J.W. Hooper, R.O. Chester, *Software Reuse Guidelines and Methods*, Plenum Press, New York, 1991.
- [25] I. Jacobson, M. Griss, P. Jonsson, *Software Reuse: Architecture Process and Organization for Business Success*, Addison-Wesley, Reading, MA, 1997.
- [26] J. Jiang, G. Klein, Risks to different aspects of system success, *Information and Management* 36 (5), 1999, pp. 263–272.
- [27] R.E. Johnson, B. Foote, Designing reusable classes, in: R. Prieto-Diaz, G. Arango (Eds.), *Domain Analysis and Software Systems Modeling*, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 138–147.
- [28] R. Joos, Software reuse at Motorola, *IEEE Software* 11, 1994, pp. 42–47.
- [29] J. Karimi, An asset-based development approach to software reusability, *MIS Quarterly* (June) 14 (2), 1990, pp. 179–198.
- [30] Y. Kim, E.A. Storr, Software reuse: survey and research directions, *Journal of Management Information Systems* 14 (4), 1998, pp. 113.
- [31] L.J. Kirsch, Portfolios of control modes and IS project management, *Information Systems Research* 8 (3), 1997, pp. 215–239.
- [32] W. Kozaczynski, Granularity of objects for effective reuse, in: *Proceedings of the Third International Conference on Software Reuse*, Rio de Janeiro, Brazil, 1–4 November 1994, pp. 210–211.
- [33] N.A. Maiden, A.G. Sutcliffe, People-oriented software reuse: the very thought, in: *Proceedings of the Second International Workshop on Software Reusability (IWSR-2)*, Los Alamitos, CA, 1993, pp. 176–185.
- [34] M.I. Markus, D. Robey, Information technology and organizational change: causal structure in theory and research, *Management Science* 15 (5), 1988, pp. 583–598.
- [35] R. McCain, Reusable software component construction: a product-oriented paradigm, in: R. Prieto-Diaz, G. Arango (Eds.), *Domain Analysis and Software Systems Modeling*, IEEE Computer Society Press, Los Alamitos, CA, 1991, pp. 132–146.
- [36] M.B. Miles, A.M. Huberman, *Qualitative Data Analysis: A Sourcebook of New Methods*, Sage, Newbury Park, CA, 1984.
- [37] H. Mili, F. Mili, A. Mili, Reusing software: issues and research directions, *IEEE Transactions on Software Engineering* 21 (6), 1995, pp. 528–562.
- [38] D.L. Nelson, Individual adjustment to information-driven technologies: a critical review, *MIS Quarterly* 14 (1), 1990, pp. 79–99.
- [39] W.J. Orlikowski, J.J. Baroudi, Studying information technology in organizations: research approaches and assumptions, *Information Systems Research* 2 (1), 1991, pp. 1–28.
- [40] W.J. Orlikowski, The duality of technology: rethinking the concept of technology in organizations, *Organization Science* 3 (3), 1992, pp. 398–427.
- [41] J.S. Poulin, *Measuring Software Reuse: Principles, Practices, and Economic Models*, Addison-Wesley, Reading, MA, 1997.
- [42] J.S. Poulin, J.M. Caruso, A reuse metrics and return on investment model, in: *Proceedings of the Third International Conference on Software Reuse*, Los Alamitos, CA, 1993, pp. 152–166.
- [43] R. Prieto-Diaz, G. Arango, *Domain Analysis and Software Systems Modeling*, IEEE Computer Society Press, Los Alamitos, CA, 1991.
- [44] A. Rai, G.S. Howard, An organizational context for CASE innovation, *Information Resources Management Journal* 6 (3), 1993, pp. 21–34.
- [45] A. Rai, A. Hindi, The effects of development process modeling and task uncertainty on development quality performance, *Information and Management* 37 (6), 2000, pp. 335–346.
- [46] D. Robey, M. Boudreau, Accounting for the contradictory organizational consequences of information technology: theoretical directions and methodological implications, *Information Systems Research* 10 (2), 1999, pp. 167–186.
- [47] M.A. Rothenberger, K.J. Dooley, A performance measure for software reuse projects, *Decision Sciences* 30 (4), 1999, pp. 1131–1153.
- [48] M. A. Rothenberger, J.C. Hershauer, A software reuse measure: monitoring an enterprise-level model driven devel-

- opment process, *Information and Management* 35 (5), 1999, pp. 283–294.
- [49] A Reuse-Based Software Development Methodology, CMU/SEI-92-SR-004, Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, PA, 1992.
  - [50] M.A. Simos, The domain-oriented software life cycle: towards an extended process model for reusability, in: G. Booch, L. Williams (Eds.), *Proceedings of the Workshop on Software Reuse*, Software Productivity Consortium, Boulder, CO, 1987, pp. 42–46.
  - [51] Reuse Adoption Guidebook, version 02.00.05, SPC-92051-CMC, Software Productivity Consortium (SPC), Herndon, VA, 1993.
  - [52] Software Technology for Adaptable Reliable Systems (STARS), 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.
  - [53] Software Technology for Adaptable Reliable Systems (STARS), Organizational Domain Modeling (ODM) Guidebook, version 2.0, STARS-VC-A025/001/00, Lockheed Martin Tactical Defense Systems, Reston, VA, 1996.
  - [54] Software Technology for Adaptable Reliable Systems (STARS), Conceptual Framework for Reuse Processes, STARS Technical Report, STARS-VC-A018/001/00, Reston, VA, October 1993.
  - [55] A. Strauss, J. Corbin, *Basics of Qualitative Research: Grounded Theory, Procedures, and Techniques*, Sage, Newbury Park, CA, 1990.
  - [56] S.W. Sussman, P.J. Guinan, Antidotes for high complexity and ambiguity in software development, *Information and Management* 36 (1), 1999, pp. 23–36.
  - [57] W. Tracz, The three cons of software reuse, in: *Proceedings of the Third Annual Workshop on Tools and Environments for Reuse*, Syracuse, NY, 1990.
  - [58] J. Withey, *Investment Analysis of Software Assets for Product Lines*, CMU/SEI-96-TR-010, Software Engineering Institute (SEI), Carnegie Mellon University, Pittsburgh, PA, 1996.
  - [59] R.K. Yin, *Case Study Research: Design and Methods*, Sage, Beverly Hills, CA, 1994.

**Karma Sherif** is an assistant professor in the Department of Information Systems and Quantitative Science at Texas Tech University. She teaches object-oriented programming. Her research interests are in software reuse technology and knowledge management systems.

**Ajay Vinze** is the Earl and Gladys Davis distinguished professor at the School of Accountancy and Information Management at Arizona State University. He teaches courses on intelligent decision systems and research methods. His research interests focus on computer-supported collaborative work, business intelligence, and cross-cultural applications of information technology.