August 2006

Managing the Complementarity of Knowledge Integration and Process Formalization for Systems Development Performance

Ravi Patnayakuni
University of Alabama, r.patnayakuni@email.uah.edu

Cynthia P. Ruppel
University of Alabama, ruppelc@email.uah.edu

Follow this and additional works at: https://aisel.aisnet.org/jais

Managing the Complementarity of Knowledge Integration and Process Formalization for Systems Development Performance [1]

Ravi Patnayakuni
College of Administrative Sciences
Department of Economics & Information Systems
University of Alabama in Huntsville, Huntsville, AL 35899
r.patnayakuni@email.uah.edu

Cynthia P. Ruppel
College of Administrative Sciences
Department of Economics & Information Systems
University of Alabama in Huntsville, Huntsville, AL 35899
ruppelc@email.uah.edu

Arun Rai
Center for Process Innovation &
Department of Computer Information Systems
Robinson College of Business
Georgia State University, Atlanta. GA. 30303
arunrai@gsu.edu

Abstract

Systems development processes have received significant negative publicity due to failed projects, often at large costs, and performance issues that continue to plague IS managers. This study complements existing systems development research by proposing a knowledge management perspective for managing tacit and explicit knowledge in the systems development process. Specifically, it proposes that collaborative exchange and integration of explicit knowledge across phases of the development process positively influence the performance of systems development. It also suggests that process formalization not only directly impacts development performance but also moderates the performance effects of the knowledge integration factors. Data for the empirical study were collected from 60 organizations that are part of a user group for one of the world's largest software development tool vendors.

---

[1] Jeffrey Parsons was the accepting senior editor. This paper was submitted on March 1, 2005, and went through 3 revisions.

Empirical results provide strong evidence of the importance of supporting tacit and explicit knowledge processes in systems development as well as process formalization. The findings suggest that: (i) collaborative exchange among IS employees that integrates their tacit knowledge positively impacts development performance, (ii) explicit knowledge integration in development artifacts across different phases of the systems development process positively impacts development performance, (iii) formalization of processes that establishes routines and discipline yields performance gains, and (iv) the performance effects of both collaborative exchange and explicit knowledge integration are moderated by the formalization of the process. These results have implications for how both tacit and explicit knowledge integration can be managed during systems development, and how formalization of processes complements their relationship with development performance.

Keywords: Knowledge exchange, explicit and tacit knowledge, collaboration, process formalization, software development performance

Introduction

Managing systems development projects so that the desired functionality is delivered in time and on budget continues to elude organizations (Tiwana and Keil, 2004; Wallace et al., 2004). It is estimated that U.S. corporations spent approximately $1 trillion on underperforming IT projects during the period 1997-2001, accounting for nearly 40% of total IT expenditure (Benko and McFarlan, 2003). Moreover, about 75% of IS projects are late, over budget, do not deliver core functionality, or are cancelled outright (Glen, 2006). Similarly, on average, one-third of a software development project's budget is spent on fixing defects that originated from faulty requirements (Pratt, 2006). Addressing these performance problems related to systems development becomes even more important due to increased organizational dependence on information systems for mission-critical activities and the magnitude of potential losses associated with poor systems quality. In today's business environment, which is fast-paced and rapidly changing, organizations must be able to respond to change with information systems that are delivered or modified with agility.

Over the past two decades, research and practice in systems development have been dominated by the view that the application of engineering principles will lead to a more manageable, predictable, and disciplined systems development process with consistent performance outcomes. Tools support Computer-Aided Software Engineering (CASE), new development methodologies, and new modeling techniques and frameworks like the Capability Maturity Model (CMM) attempt to formalize the development process and improve control to provide better outcomes in terms of quality, time, and budget. More recently, in response to the rapidly changing business and technical environment, there is an increased emphasis on imperatives like responsiveness, time to market, and programming skills (Baskerville and Pries-Heje, 2001). To complement the engineering perspective, researchers (Purvis et al., 2001; Walz et al., 1993) have suggested that systems development should be viewed as a knowledge-intensive activity and that the systems development process and its outcomes should be examined from a knowledge management perspective. This is the approach we adopt in this study. Rather than focusing on a specific genre of tools, techniques, and methods, we focus on managing tacit and explicit knowledge in the systems development process with a view toward realizing better outcomes.

Our investigation is informed by the rich literature on knowledge management and process formalization. Based on a review of research in knowledge management systems, Alavi and Leidner (2001) develop a framework of knowledge management processes that views organizations as social collectives and "knowledge systems." Knowledge management, according to Kwan and Balasubramanian (2003, p. 204), "…involves setting up an environment that allows workers in organizations to create, capture, share and leverage knowledge to improve performance." These perspectives provide the basis for developing a complementary approach to address systems development problems based on knowledge management. In fact, the knowledge management perspective has been effectively applied to address performance problems in related contexts, such as product development. By framing new product development as knowledge-intensive work, it was found that knowledge about customers, suppliers, and internal capabilities, as well as the sharing of this knowledge, results in improved performance in new product development teams (Hong et al., 2004). Similarly, systems development is considered to be a knowledge intensive process (Hoegl et al., 2003; Purvis et al., 2001; Walz et al., 1993). More recently, Patnayakuni et al. (2006) build on the concept of 'boundary objects' proposed by Carlile (2002) to examine how formal and informal organizational integrative practices shape the development of artifacts that can be ported across knowledge boundaries and how these artifacts impact performance. Yet, there has been limited theoretical and empirical investigation of how systems development problems can be addressed by supporting the tacit and explicit integration of knowledge or, importantly, of how knowledge management practices can be reconciled with other process management practices, specifically disciplined and formalized process management.

Applying a knowledge perspective, we argue that collaborative exchange for tacit and explicit knowledge integration in the development artifacts that are generated and used across the systems development process is likely to lead to improved performance in systems development. Additionally, we argue that the extent to which the development process is formalized acts as a quasi-moderator on the relationship between these knowledge integration capabilities and performance of the systems development process. We believe this is because it has a direct effect on systems development performance and it also moderates the relationship between knowledge integration capabilities and systems development performance.

We employed a survey of systems development managers, who were associated with the user group of a major software vendor, to collect data on systems development practices. This study contributes to existing literature by indicating how knowledge integration capabilities and process formalization impact systems development performance, explicating over 40% of the variance in systems development performance. Specifically, it demonstrates that: (1) collaborative exchange among employees contributes to increased performance by integrating their tacit knowledge, (2) integration of explicit knowledge in development artifacts contributes to increased performance by streamlining the exchange of knowledge across phases of the development process, (3) formalized processes enhance performance through institutionalization of effective routines and practices, and (4) the impact of the collaborative exchange and explicit knowledge integration on performance is moderated by the formalization of the systems development process. While formalization appears to dampen the performance gains from explicit knowledge integration, it likewise appears to amplify the performance gains that accrue from collaborative exchange. Collectively, the findings support the adoption of a knowledge management approach in tandem with process formalization to address

systems development performance problems. IS managers should look not only at development practices (e.g., methodologies, methods and technology) but also at the organizational practices that support or constrain the integration of tacit and explicit knowledge across the development process.

The remainder of the paper is organized as follows: Initially, we develop our theoretical framework, followed by a specification of our research model and hypotheses. We then describe the empirical methodology, including the research method, survey instrument development, and sampling strategy. Subsequently, we present the results of our analysis on validation of measures and hypothesis testing. We conclude by interpreting these results and deriving implications for theory, practice, future research, and pedagogy.

A Process View of Knowledge

The knowledge-based view of the organization argues that a firm should be viewed as a social community that specializes in the creation and transfer of knowledge (Kogut and Zander, 1996). This view of organizations is based on the premise that knowledge is a central resource and that it is the heterogeneous stocks and flows of knowledge in a firm that provide it with unique resources and performance capabilities.

Researchers investigating knowledge-related phenomena must tackle the dual tasks of conceptualizing knowledge and operationalizing its measurement. A common approach is one that draws a distinction between tacit and explicit knowledge. Explicit knowledge can be expressed in some symbolic form, making it easier to communicate and transfer than tacit knowledge, which is abstract and difficult to formalize (Alavi and Leidner, 2001; Nonaka and Konno, 1998). Tsoukas (1996) observes that the taxonomic distinction between tacit and explicit knowledge creates an artificial dichotomy that fails to recognize that "tacit and explicit knowledge are mutually constituted" (p. 14) and that tacit knowledge is an integral part of all knowledge. Such views are predicated on the proposition that knowledge is something that exists only in the minds of people and is shaped by experience as well as the situation at hand, thus making it difficult to study knowledge itself. The abstract and metaphysical nature of knowledge makes it difficult to approach it only as an object that can be stored and manipulated and is likely to lead to different (if not contradictory) opinions on how knowledge should be managed in organizations (Alavi and Leidner, 2001).

Thus, a process view of knowledge has been suggested as an alternative to the perspective of knowledge as an object (Alavi and Leidner, 2001; Davenport et al., 1998; Orlikowski, 2002; Tsoukas, 1996). Knowledge as a process implies that knowledge is about simultaneously knowing and acting; that it is an "ongoing social accomplishment, constituted and reconstituted as actors engage the world in practice" (Orlikowski, 2002, pg. 249). Viewed as a process, the focus is on flows of knowledge, i.e., its generation and deployment (Alavi and Leidner, 2001). Additionally, this view implicitly makes a distinction between tacit and explicit knowledge as well as espouses the idea that distinct mechanisms are likely to be effective at integrating each of these types of knowledge.

We adopt the process view of knowledge to examine how the tacit knowledge of IS developers and the explicit knowledge generated in the systems development process

are integrated, as well as how these two modalities of integration impact performance of the process. Moreover, we examine how process management, specifically, formalization of the development process, complements these two modes of knowledge integration. Figure 1 depicts our research framework. The core premise is that an organization's systems development performance is a function of its ability to integrate knowledge throughout the systems development process, and to use knowledge assets systematically. Furthermore, we propose that process formalization is a quasi-moderator (Sharma, et al., 1981) in that it has a direct effect on systems development performance and also moderates the relationship between knowledge integration and systems development performance.

Figure 1. Theoretical Framework
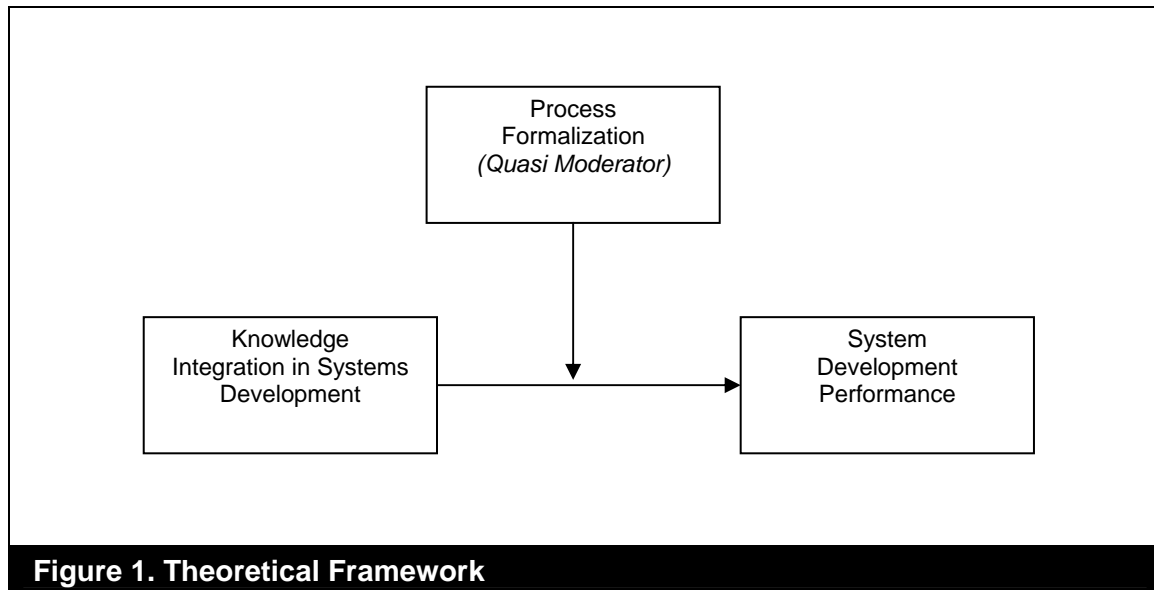
Supporting Knowledge Integration in Systems Development: Constructs and Relationships

Knowledge management is concerned with how people, processes, and technology-enabling components can enhance knowledge resources, and thus, the performance of a process, organizational unit, or firm (Hawryszkiewycz, 2005; Hansen et al., 1999; Kwan and Balasuramanian, 2003; Maier and Remus, 2002). As stated, some knowledge management taxonomies draw a distinction between tacit and explicit knowledge (Alavi and Leidner, 2001), and it has been further suggested that they be viewed both collectively and as complementary (Alavi and Leidner, 2001; Orlikowski, 2002), as tacit knowledge provides the required background to interpret and give meaning to explicit knowledge (Polanyi, 1966). Thus, individuals with a high degree of shared understanding are more likely to be effective in exchanging and integrating knowledge. Similarly, several authors have emphasized the interaction between the individual and the collective as an important aspect of organizational knowledge integration (Polanyi, 1966; Tsoukas, 1996; Brown and Duguid, 1998; Tuomi, 1999; Alavi, 2000). Accordingly, we consider the integration of both tacit and explicit knowledge in our investigation of systems development performance.

The literature suggests that both process and outcome measures of performance can be impacted by knowledge management (Alavi and Leidner, 2001; Davenport et al., 1998;

Orlikowski, 2002). Thus, we consider in our investigation: (1) outcome performance, defined as the degree to which systems developed by the IS department meet requirements in terms of functionality, quality, and user satisfaction and (2) process performance, defined as the degree to which systems developed by the IS department meet productivity and efficiency objectives.

To improve systems development performance, researchers have focused on a variety of approaches, such as development methodologies (Vessey and Conger, 1993; Iivari, et al., 2001), design paradigms (Alter, 2001; Baskerville and Pries-Heje, 2001), CASE tools (Rai and Patnayakuni, 1996; Orlikowski 1993), risk management approaches (Keil et al., 1998; Schmidt, et al., 2001), coordination strategies (Nidumolu and Subramani, 2004), process improvement frameworks (Paulk et al., 1995), and behavioral factors that can improve the development process (Kirsch, 1996). These approaches focus primarily on controlling and managing the development process using tools, techniques, and methods that will result in improved systems development performance that is consistently replicable across projects. While past research has provided several useful insights, systems development projects continue to perform poorly in terms of meeting requirements, going over budget, and being delayed, if not failing completely (Keil et al., 2000).

Given the inherently knowledge-intensive nature of the development process, and its continued performance pitfalls, we assert that successful systems development requires the integration of specialized knowledge that is dispersed across roles (e.g., analysts, programmers, line managers, users) or phases of the process (e.g., requirement determinations and design). This implies that capabilities to exchange tacit and explicit knowledge across specialized roles as well as phases of the process will enhance development performance.

Collaborative Exchange: Integrating Tacit Knowledge

Nonaka and Konno (1998) identify socialization, externalization, combination, and internalization as different modes for knowledge creation and application. Externalization and internalization refer to the interaction and conversion of knowledge at the individual level – integration of knowledge by articulating tacit knowledge and creation of tacit knowledge by working and interacting with others. Additionally, combination can be achieved by filtering, merging, organizing, synthesizing, and summarizing codified explicit knowledge. From an organizational perspective, socialization, where tacit knowledge is integrated with the tacit knowledge of other employees through social interactions and shared experiences, is perhaps the most significant. Alavi (2000) states, "Coherent and synergistic organizational knowledge is generated through collaboration, interactions, and relations among individuals" (pg 19). While knowledge is "owned" and "enacted" in the minds of individual employees, the integration of this knowledge to a collective level is both necessary and fundamental (Okhuysen and Eisenhardt, 2002). Similarly, Nonaka and Konno (1998) emphasize the role of interaction and note that knowledge generation is enabled by a rich space or *ba* that promotes interaction. In a *ba*, individuals share their tacit knowledge, learn from each other's tacit knowledge, collectively apply their tacit knowledge to interpret explicit knowledge, and further evolve the collective knowledge base of the organization.

In general, informal exchanges routinely occur in organizations. Employees often 'walk down the corridor' and/or ask a 'quick question' of a co-worker to resolve problems or

exchange information. Related to the emphasis on socialization by Nonaka and others (Nonaka, 1994; Nonaka and Konno, 1998; Nonaka and Toyama, 2003), Davenport and Prusak (2000) argue that the spontaneous, unstructured exchange of knowledge is a critical ingredient for creating and integrating knowledge in organizations. Similarly, other researchers have discussed the importance of informal communication for feedback and socialization (Sarbaugh-Thompson and Feldman, 1998).

Given the importance of socialization, the systems development process should involve a significant amount of communication and shared contextual experience among its stakeholders to facilitate knowledge integration (DeFranco-Tommarillo and Deek, 2004; Walz et al., 1993). Interaction among users and customers with the development team is considered very important (Doll and Torkzadeh, 1989; McKeen and Guimaraes, 1997). A study of 145 systems development projects found that team members collaborate to share information and skills necessary to complete the project, particularly for projects that are innovative or complex (Hoegl et al., 2003).

In summary, collaboration should be designed as part of the development process to promote exchange and integration of tacit knowledge among key constituents, including developers and users. Within the broader rubric of socialization, we focus specifically on informal communication in the information systems department. Accordingly, we define collaborative exchange as the degree to which informal communication exists among participants in the development process. This collaboration should provide the contextual specificity necessary to meaningfully create and share knowledge. For example, it should facilitate developers' integration of their technical knowledge with the application domain knowledge of users. Similarly, different user groups, defined by product groups, functional membership, or location, should be able to more effectively share their perspectives on the application domain with others. Additionally, developers should be able to share their technical knowledge with other developers. The suggested permutations and combinations of integrating technically and contextually-specific knowledge between and among developers and user groups expands the knowledge base for the development process and promotes common understanding.

Given our discussion thus far, we hypothesize:

H1: The use of collaborative exchange for tacit knowledge integration positively impacts systems development performance.

Explicit Knowledge Integration

Knowledge created in the organization can be used to develop organizational capabilities by codifying it in processes (Grant, 1996). In fact, to be easily communicated, transferred, and applied, tacit knowledge should be codified. Organizations, therefore, need to have mechanisms to promote the capture, representation, and application of tacit knowledge. While collaborative exchange enhances individual learning by expanding participants' individual tacit knowledge through the process of sharing and transfer, individuals also generate explicit knowledge in the form of development artifacts (for example, requirements, use cases, E-R models etc.) that integrate knowledge across the application and technical domain.

Typically, processes—and systems development processes in particular—consist of tasks that are separated by time and location and dispersed across stakeholders and

roles. Knowledge associated with different phases of development, such as requirements specifications, logical models, and physical design, have to be integrated to achieve process-wide consistency and retention of critical information from phase to phase. Consistent with this view, Chen (2005) suggests that it is necessary for knowledge to be accessible to others and subject to application, change, and adaptation by others in the organization. Similarly, Hoopes and Postrel (1999) make a distinction between the act of sharing knowledge (as discussed above in collaboration) and shared knowledge, where shared knowledge is defined as the actual "facts, concepts and propositions which are understood simultaneously by multiple agents (pg 863)." They found that gaps in the understanding and interpretation of information resulted in significant excess costs to the organization in product development.

More recently, the role of explicit knowledge integration has been investigated by Patnayakuni et al. (2006). Based on survey data collected from IS departments in 119 organizations, they found that the integration of knowledge across development artifacts leads to greater systems development performance. Thus, theoretical arguments and past empirical evidence suggest that in addition to developing a robust logic to relate tasks and define their interfaces, it is critical to establish a shared language for the consistent and meaningful flow of information across the development process. This depends on the degree to which a common base of explicit knowledge has been captured, shared, and formalized, so as to enable consistent and meaningful interpretation of artifacts across the development process.

In this study, we focus on explicit knowledge integration, the degree to which semantic consistency, accessibility, and the ability to share information is achieved across stages of the development process. In this study, explicit knowledge integration represents the integration of knowledge in artifacts that are generated in the system development process. Systems development tasks create and manipulate artifacts that should exhibit integrity for effective systems development. To achieve such integrity, the information content of development artifacts created in the initial stages (such as requirements documents) needs to be preserved for access and use later in the process (such as programming and testing). Thus, we hypothesize:

H2: Explicit knowledge integration across phases of the development process positively impacts systems development performance.

Process Formalization

We are interested in the role of formalization in influencing the relationship between knowledge integration capabilities and systems development outcomes. It has been suggested that knowledge integration may be either constrained or enabled by the structural characteristics of organizations (Alavi and Leidner, 2001; Davenport and Prusak, 2000; Nonaka and Konno, 1998). Higher levels of formalization are marked by minimal redundancy of tasks, de-emphasis of collaboration, and a focus on hierarchical control and task efficiency. In contrast, knowledge creation is associated with institutionalization of decision structures and work processes that enable collaboration and cross-fertilization of individual employee knowledge (Melcher et al., 1990). Thus, while formalized structures can enhance performance, they can also dampen the effects of certain other causal pathways that lead to performance gains.

In essence, formalized structures embed knowledge into stringent routines, institutionalize behaviors (including behaviors related to the creation and use of knowledge) and suppress collaboration. Along these lines, Dougherty (1992) argued that established routines and rules, characteristic of formal structures, create barriers to knowledge integration. Others have pointed to the negative consequences of formal structure for knowledge transfer as they restrict channels for knowledge flows and inhibit interactions (Okhuysen and Eisenhardt, 2002). Similarly, formal structures can constrain the adaptive use of knowledge, given changes in requirements as the project progresses. Accordingly, we posit that process formalization will have a quasi-moderating effect on systems development performance: higher levels of formalization will positively impact systems development performance but also negatively moderate the relationship between knowledge integration capabilities and systems development performance.

H3: Process formalization will positively influence systems development performance.

H4: Process formalization will negatively influence the relationship between collaborative exchange and development performance.

H5: Process formalization will negatively influence the relationship between explicit knowledge integration and development performance.

Control Variables

We defined IS department size and tool portfolio age as control variables. Size has been used as a control variable in several studies focusing on knowledge-based issues in assimilating software process innovations and in other IT adoption and diffusion studies (Fichman and Kemerer, 1997). Using length of experience with a tool as a control variable reduces confounding effects due to tool-focused learning or issues associated with stabilizing the technology infrastructure. The research variables and their hypothesized relationships are represented in Figure 2.
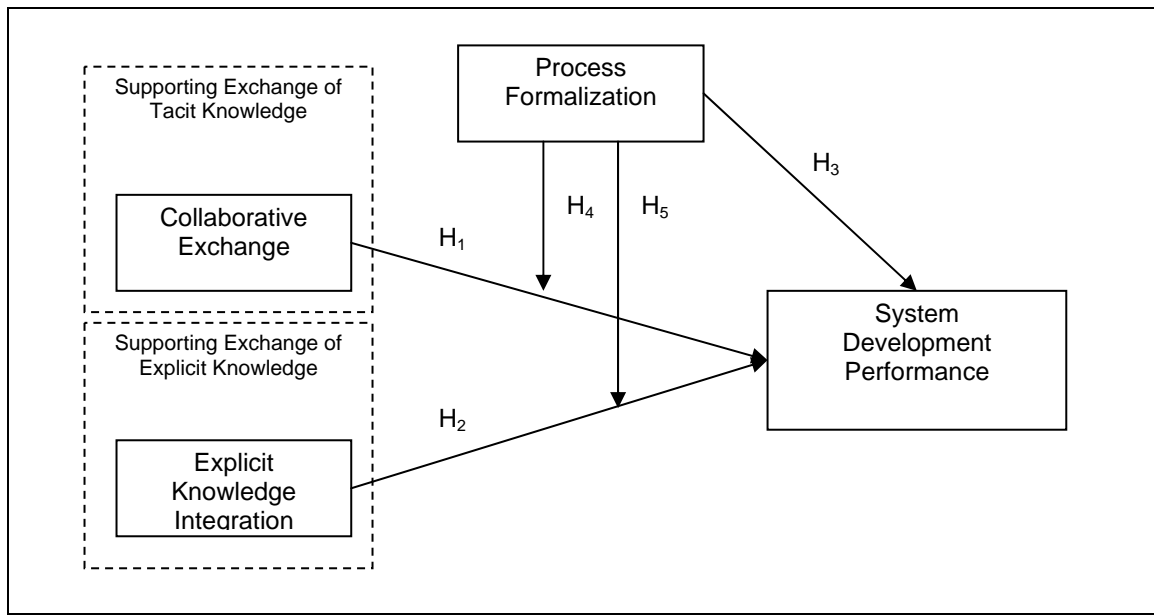

Figure 2. Theoretical Model

Methodology

Data Collection

Survey data were collected from the user group of a large independent software vendor's CASE tools. The vendor organization sponsored the data collection, as its managers viewed the study as an opportunity to learn more about their clients' development practices. Since the study focuses on theory building and an exploratory test of the theoretical model, collecting data from organizations using the development tools of one vendor provides for homogeneity and control in the technology platform and vendor-organization interfaces.

The vendor organization distributed the survey directly to its key contact in each respondent organization, soliciting a response or requesting the contact to forward the survey to the member of the organization most qualified to respond. The survey sponsor characterized the user group as active and well-informed about industry development practices.   Although the researchers did not have access to the direct contact information of potential respondents, the high response rate and some self-report data suggest that the survey was administered to the target profile by the software vendor. In total, the survey instrument was distributed to approximately 100 user organizations, of which 60 completed the survey for a response rate of 60%.   All respondents were directly involved with systems development in their organizations and, in addition, were directing the implementation of the software vendor's CASE tools. Self-report data, where provided, revealed that respondent titles ranged from senior systems analyst to VP of systems development.

Scales and Measurement Properties

We developed the survey instrument using a three-phase instrument development process under the guidelines suggested by Straub (1989) and Sethi and King (1991). Since established measures were unavailable for the explicit knowledge integration construct, in the first phase, we reviewed the relevant literature (Mi and Scacchi, 1992; Hoopes and Postrel, 1999; Chen, 2005) to develop a list of items for this construct. This scale is also used by Patnayakuni et al. (2006) in a separate national survey of systems development manager on software development practices. We based multi-item measures for other constructs on prior research that had examined them. Specifically, informal communication was informed by Davenport et al. (1998), Nilkanta and Scammell (1990), and Zmud (1982); process formalization was informed by Zmud (1982) and Davenport and Short (1990); and systems development performance by Baroudi et al. (1986), Finlay and Mitchell (1994), and Ravichandran and Rai (2000)**.** We measured the two control variables using single item measures. We measured IS department size by the number of full-time-equivalent employees in the IS department and prior experience with tools by the number of years the current development tool portfolio was in use in the organization.

All constructs were operationalized at the level of the development process, consistent with the proposed model's unit of analysis. This is the most appropriate level of analysis in view of the constructs specified in the model, where collaborative exchange captures the extent of informal communication in the process across projects and in the IS department, in general.   Similarly, data on formalization and explicit knowledge

integration are captured at the level of the IS development process and represent systemic capabilities applied across systems development projects.

In the second phase of instrument development, we conducted interviews with six senior IS managers responsible for managing systems development in their respective organizations. These interviews focused on the development practices and the implementation of CASE tools in these six organizations. The interviews served as useful input into the questionnaire development process. Finally, we pilot-tested the instrument with four senior IS executives from different organizations, four faculty members, and two doctoral students conducting research on the management of systems development. Comments received focused on item wording and formatting of the instrument, which we addressed prior to collecting data from the user group.

Table 1 shows the constructs and measurement items used in the survey instrument. We used seven-point Likert scales for items associated with collaborative exchange, process formalization, explicit knowledge integration, process performance, and outcome performance.

Analysis of the data was conducted using partial least squares (PLS) using PLS Graph v3.0 to evaluate the measurement properties and structural relationships specified in the research model. PLS is considered an appropriate analytical approach for the study as it allows: (1) modeling of latent constructs as either formative or reflective and (2) assessment of psychometric properties of the constructs (the measurement model) within its theoretical context (the structural model). We conducted the analyses in two stages. First, we tested the measurement model to ensure that the constructs had sufficient psychometric validity, then addressed the structural model in which the hypotheses were tested.

In considering whether our constructs should be modeled as reflective or formative, we drew on Jarvis et al. (2003). In their review of measurement modeling in marketing and consumer research, they suggest that researchers often misspecify formative constructs as reflective and develop guidelines to avoid such errors. They note that the decision to model a construct as formative or reflective should be based on the following four criteria: (1) direction of causality from construct to indicators, (2) interchangeability of indicators, (3) co-variation among indicators, (4) nomological net of construct indicators (Jarvis et al., 2003). We modeled constructs as formative if the direction of causality is from indicators to constructs, indicators need not be interchangeable, indicators need not covary, and the nomological net of indicators can differ. We modeled them as reflective if the opposite conditions are applicable.

By applying Jarvis et al.'s (2003) guidelines, all constructs in the model are modeled as formative. In the case of the systems development performance construct, outcome performance need not necessarily be accompanied by higher levels of process performance in terms of adherence to budget and schedules. Similarly, the five-item explicit knowledge integration construct measures consistency of development artifacts that do not necessarily have to co-vary even though they could be mutually reinforcing. The same logic applies to collaborative exchanges where interaction among developers may not necessarily coincide with similar levels of interaction with other stakeholders. Finally for process formalization where automation need not necessarily co-vary with formalized and structured process is also modeled as formative.

Table 1: Item Loadings and Reliability

Item		Factor Loading	Composite Reliability
Collaborative Exchange			
CE1	There is extensive informal communication among IS employees at the same level.	.82	.884
CE2	There is extensive information communication between IS employees and employees at the same level in other departments.	.80	
CE3	Developers interact with each other on a routine Basis.	.87	
Explicit Knowledge Integration			
EKI1	Data generated during a particular task/phase of systems development is easily accessed by related tasks/phases	.84	.845
EKI2	Modifications made to development information in a particular task/phase are communicated to related tasks/phases	.68	
EKI3	Development information is easily portable from one development task/phase to other tasks/phases	.69	
EKI4	Logical models remain consistent across different development tasks/phases	.60	
EKI5	No semantic information is lost in moving from one task/phase of development to another	.63	
Process Formalization			
PF1	Routine systems development tasks are automated	.78	.848
PF2	Task in projects have been formalized and structured as routine	.90	
Process Performance			
PP1	Projects finish within budgets	.86	.855
PP2	Projects finish on schedule	.88	
PP3	Productivity of our development staff is high compared to other IS organizations in similar Environments	.74	
PP4	Users are dissatisfied with the lead time for systems delivery (reverse coded)	.64	
Outcome Performance			
OP1	Users are satisfied with developed systems	.82	.884
OP2	Systems that have been developed have high Reliability	.83	
OP3	Fixing bugs and other rework account for a significant proportion of our development effort (reverse coded)	.87	
OP4	Users are satisfied with the overall quality of developed systems	.68	

Notes: As with reflective constructs, formative constructs can exhibit convergent validity and internal consistency, as is the case here. However, given the errors in the specification of measurement models noted in recent reviews of the literature (Jarvis et al, 2003), it is very important to point out that these properties are not necessary conditions for formative constructs. However, when formative constructs exhibit convergent validity and composite validity, a unit weight mean score can be used (Rozeboom, 1979)

Table 2: Discriminant Validity and Descriptive Statistics

	Mean (Standard Deviation)	CE	EKI	PF	PP	OP
Collaborative Exchange (CE)	5.17 (1.14)	**.847**				
Explicit Knowledge Integration (EKI)	5.40 (.964)	.017	**.724**			
Process Formalization (PF)	4.06 (1.32)	.368	.228	**.858**		
Process Performance (PP)	4.22 (.88)	.347	.239	.117	**.775**	
Outcome Performance (OP)	4.63 (.79)	.325	.359	.345	.358	**.811**

Notes: Diagonal elements are the square root of average variance extracted from the measurement items for each construct.

Table 3: Item-to-Construct Correlations

	CE	EKI	PF	PP	OP
CE1	**.872**	-.061	.340	.290	.192
CE2	**.803**	.025	.304	.305	.266
CE3	**.863**	.099	.281	.283	.394
EKI1	.069	**.794**	.187	.435	.204
EKI2	-.139	**.773**	.113	.108	.296
EKI3	.032	**.665**	.347	.136	.339
EKI4	.023	**.708**	.133	.154	.185
EKI5	.114	**.673**	.103	.150	.264
PF1	.343	.132	**.865**	.116	.249
PF2	.289	.262	**.857**	.086	.346
PP1	.351	.278	.187	**.821**	.367
PP2	.228	.307	.181	**.827**	.329
PP3	.223	.091	.252	**.713**	.310
PP4	.023	.111	.396	**.789**	.158
OP1	.329	.311	.314	.474	**.828**
OP2	.386	.346	.338	.512	**.669**
OP3	.249	.119	.199	.521	**.898**
OP4	.281	.262	.342	.404	**.835**

We report the results of the measurement analysis in Table 1. It should be noted that convergent validity and internal consistency are not requisite criteria for formative measures (Jarvis et al., 2003), though these constructs may exhibit these properties commonly associated with and expected of reflective constructs. However, these constructs should exhibit discriminant validity. To assess the measurement properties of our scales, we comprehensively evaluated the nature of convergence, discrimination, and reliability of indicators.

As can be observed from Table 1, the composite reliability of all constructs is 0.7 or higher. We obtain further evidence of discriminant and convergent validity from an examination of the diagonal elements of the correlation matrix (which represent the square root of AVE) in Table 2. All diagonal elements exceed the off-diagonal elements, which indicates acceptable discriminant validity (Chin, 1998a). Additional support for discriminant validity is obtained by examining item-to-construct correlations shown in Table 3, where items demonstrated higher correlations with their corresponding constructs rather than with other constructs. Collectively, the evidence suggests that the constructs demonstrate adequate measurement properties.

For formative indicators, which have a regression-like relationship with the latent constructs, only the weights (and not the loadings) need to be considered to evaluate the role of each indicator in the measurement of the construct (Chin, 1998a). While no minimum threshold values for indicator weights have been established, the statistical significance of the weights can be used to determine the importance of indicators in forming a latent variable. The indicators associated with systems development performance were significant, with weights of .40 (t = 3.88, p < .001) for process performance and .77 (t = 9.07, p < .001) for outcome performance. Additionally, unit weights were applied to indicators for each of the formative constructs, as this weighting scheme does not lead to a loss of power when items exhibit convergent validity (Rozeboom, 1979) as is the case here.

Structural Model

We used the partial least square (PLS) method of structural modeling to test the hypotheses. Figure 3 shows the results of our empirical test of the specified structural model.
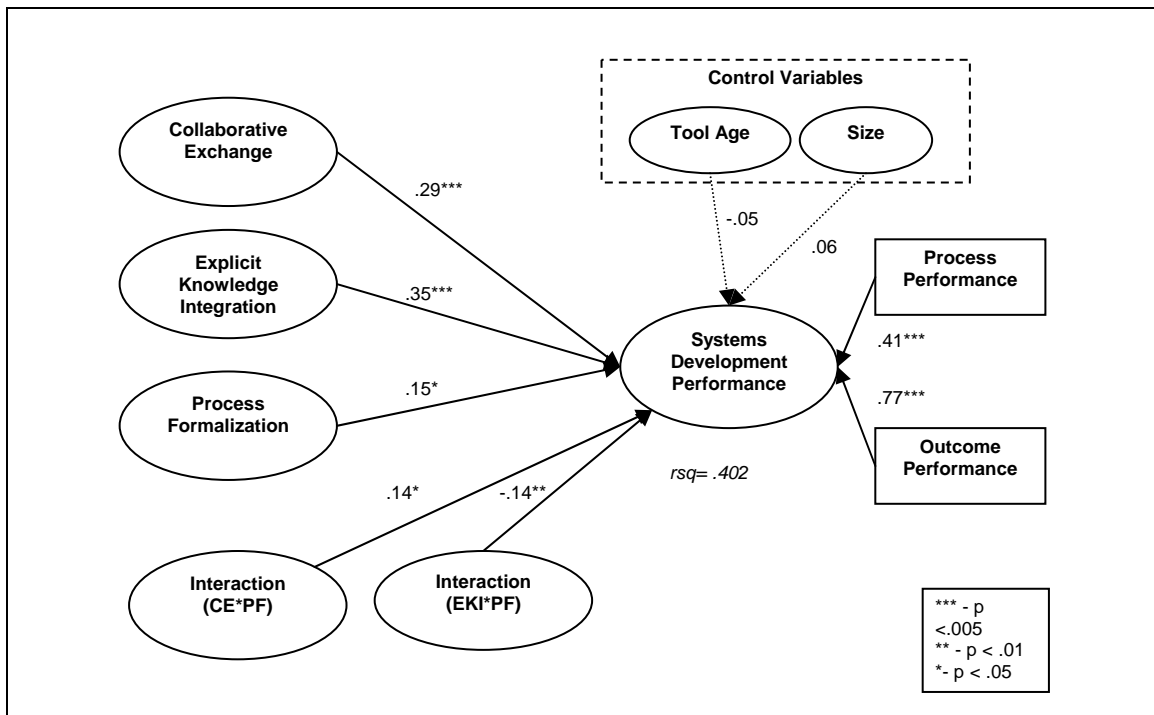
Figure 3. Results of PLS Analysis

Significance tests and estimates of confidence intervals for the path coefficients are not directly provided by the PLS method. In order to estimate the significance of path coefficients, we used a bootstrapping technique to generate 300 samples of 60 data points each. The path coefficients were re-estimated using each of these samples of observations. We used this vector of parameter estimates to compute parameter means, standard errors, path coefficient significance, indicator loadings, and indicator weights. This approach is consistent with recommended practices for estimating significance of path coefficients and indicator loadings (Löhmoller, 1984) and has been used in prior IS studies (Chin and Gopal, 1995; Compeau and Higgins, 1995; Howell and Higgins, 1990; Ravichandran and Rai, 2000).

The predictive power of the research model in PLS is assessed by examining the explained variance (R^2) for the endogenous constructs (Barclay et al., 1995, Chin, 1998b). We examined the moderating effect of process formalization as per the guidelines suggested by Chin et al. (2003). The interaction term was a paired product of all the measurement items for the individual constructs. In order to assess the extent of moderation, we analyzed the model in two stages. In the first stage, we modeled only the direct effects, including the moderating variable as a direct effect on the endogenous construct as suggested by Chin et al. (2003). This specification essentially captures the quasi-moderation effect of process formalization as it assesses its direct effect and its interaction effects with the collaborative exchange and explicit knowledge integration.

The model with only the direct effects included explained 30.4% of the variance in systems development performance, with all three direct effects significant at the .05 level. In the second stage, we introduced the interaction terms were introduced, one for the moderating effect of process formalization on the relationship between collaborative exchange and development performance and a second for the moderating effect of process formalization on the relationship between explicit knowledge integration and development performance. The addition of the moderation effects explained an additional 9.8% of the variance in development performance, for a total 40.2% of explained variance. This increase in explained variance is significant (pseudo F statistic = 13.6, p = .000), suggesting that process formalization moderates the relationship between knowledge integrative capabilities and systems development performance. Both interaction terms are significant at the .05 level and substantially add to the explained variance in development performance. The coefficient of the interaction term of process formalization and explicit knowledge integration is negative, suggesting that higher levels of process formalization inhibit the integration of explicit knowledge in development artifacts. However, the positive coefficient for the interaction of term of process formalization with collaborative exchange is positive, suggesting the higher levels of formalization complement the leveraging of tacit knowledge resident in employees. As expected, both of the control variables—tool portfolio age and IS department—size were insignificant, suggesting that they did not significantly contribute to development performance in this study.

Discussion and Implications

Knowledge management constructs are notoriously difficult to define and operationalize, largely because of complexity and the inability to directly observe most knowledge-related phenomena (Alavi and Leidner, 2001). Their relevance and explanatory power is largely determined by the nomological net in which they are embedded and investigated.

The proposed research model explained over 40% of the variance in the dependent variable (systems development performance), with support for direct effects hypotheses H1 (collaborative exchange), H2 (explicit knowledge integration), and H3 (process formalization) and one of the moderating effects (H5). Though process formalization was found to moderate the relationship between collaborative exchange and performance, the observed direction of this effect was positive and opposite to what we hypothesized (H4).

This study demonstrates the importance of managing the complementarity of knowledge integration and process formalization by showing that: (1) collaborative exchange among IS employees that integrates their tacit knowledge enhances development performance, both process and outcome, (2) semantic consistency of information across development artifacts from different phases of systems development, and their accessibility across phases, integrates explicit knowledge and enhances performance, both process and outcome, (3) process formalization, by safeguarding against common errors and institutionalizing proven routines, positively impacts development performance, and (4) the impacts of the collaborative exchange and explicit knowledge integration on development performance are moderated by the extent of formalization in the systems development process.

To elaborate on each of these findings, the effect of collaborative exchange on development performance was positive and significant. This result provides empirical support for the argument put forth by other researchers (Nonaka and Takeuchi, 1995; Purvis et al., 2001) that the presence of such communication creates the environment for habitual and practiced patterns of interaction that leverage knowledge in collective activities and facilitates knowledge integration across individual, group, and other organizational levels. In essence, collaborative exchange among stakeholders enables knowledge integration across specialized domains and enhances the outcome and process performance of systems development.

Our findings further suggest that knowledge made explicit and integrated throughout the process enhances development performance, which is consistent with Patnayakuni et al. (2006). All participants, regardless of their role, should have access to the same information to ensure that everyone is "on the same page" and that the collective action of developers across phases is effectively informed. We defined explicit knowledge integration as the degree to which semantic consistency, accessibility, and ability to share information is achieved across the stages of the development process. This result provides evidence of the importance of integrating explicit knowledge across phases of the systems development process for outcome and process performance. It also demonstrates that from an empirical standpoint, the focus on knowledge assets resident in development artifacts—which require both codification and integration of knowledge—makes the construct identified as integration of explicit knowledge more observable and measurable.

The observed positive effect of process formalization on development performance suggests that the institutionalization of routines that enforce discipline yields performance benefits. The directionality of the moderating effects of process formalization provides some interesting insights. While higher levels of formalization appear to negatively influence the relationship between explicit knowledge integration in development artifacts and development performance, they appear to strengthen the relationship between collaborative exchange and systems development performance.

This suggests that formal practices can improve the process of knowledge creation, where individual knowledge is integrated with that of other employees and into organizational routines. They provide structure to employee interactions so that individual knowledge can surface, effectively combine, and converge into shared frames of reference for new knowledge. In fact, some researchers have noted that structured activities can play a positive role in knowledge integration and transfer (Okhuysen and Eisenhardt, 2002). Similarly, Patnayakuni et al. (2006) discuss the role of formal integrative practices, such as job rotation, participative decision making, and use of teams in integrating knowledge during systems development. Thus, formal practices can be designed to promote interactions and exchange and improve knowledge transfer, as they explicitly define opportunities for employees to share know-how and know-why with others.

Our suggested approach to leverage the complementarity of knowledge integration and process formalization adds to prior work on systems development process improvement, which has focused on technical factors, coordination tools, methodology choices, control behaviors, and capability maturity as antecedents of superior development performance. Our results show that a focus on managing and supporting knowledge integration can prove beneficial in improving the performance of the systems development process, specifically by encouraging collaborative exchange and maintaining consistent and accessible development artifacts across the systems development process. Importantly, our results suggest that process formalization can yield significant performance benefits and can complement initiatives that integrate tacit knowledge. In effect, best practices to achieve consistency, safeguard against errors, and promote effective patterns of interaction among constituents can be routinized in IS development processes. However, our results imply that caution should be taken not to underutilize explicit knowledge through excessive formalization.

These identified capabilities to integrate tacit and explicit knowledge should help contain often-reported escalation of budgets and schedules associated with IS projects. Additionally, with virtual project teams that are dispersed geographically, it becomes important to deploy technology for collaborative exchange. Similarly, for projects that are off-shored, it is important to facilitate collaborative exchange for the creation and transfer of tacit knowledge between employees associated with different companies and different organizational and national cultures. Our results also suggest that projects in such contexts are likely to benefit from capabilities that support the integration of explicit knowledge generated at different phases in the process and by personnel in different organizations. It becomes important, then, to understand the mechanisms that facilitate the integration of explicit knowledge across the development process, an issue that has recently received some attention (Patnayakuni et al., 2006).

By effectively deploying tools and methods, knowledge gained from prior development projects can be captured to some extent in the form of repositories and reusable object libraries. Additionally, tools and methodologies can be leveraged to facilitate collaborative exchange through initiatives such as a project intranet, internal blogs for team members, video phones for quick meetings, and Wikis for project documents. Given the importance of knowledge integration to system development performance identified here, the relative effectiveness of each of these methods of knowledge integration for any specific development project should be explored in future research. Moreover, future research should investigate how different patterns of formal and

informal interaction shape the level and type of knowledge integration and the resultant performance effects.

From a pedagogy standpoint, this study has implications for the teaching of systems analysis and design. This study suggests classroom projects should be completed in team settings. The students should be encouraged to share knowledge and may be asked to role play various roles within the development process to ensure a variety of perspectives – not all technical as may be the case in an Information Systems classroom. Additionally, students should be encouraged to truly collaborate on projects that use collaborative exchange technologies rather than just divide project work for independent execution. For instance, they can use Wikis to create the documents that will eventually become the final document/assignment, and all group members should be encouraged to participate in evolving the document through their collaborative effort. Students can also be required to document what they learned both during and about the process. A blog, which is open to other team members, may be useful during the process. If these blogs are shared, as they should be, they can result in knowledge integration. Also, all project teams should conduct a project post-mortem to probe and document knowledge gained.

Limitations of Sampling Approach and Common Method Bias Assessment

Since this study uses survey-based perceptual data, our study is subject to the limitations inherent with this method of data collection. Additionally, it is a cross-sectional study and based on the perceptions of a single respondent. While all the respondents were actively engaged in the systems development process of the organization and leading implementation of the vendor's CASE tool, the use of a single respondent to provide perceptual survey data raises concerns of common method bias. In order to assess the extent of this problem in our data, we conducted Harmon's one-factor test (Podsakoff and Organ, 1986). In accordance with the suggested procedure, we entered all items used to measure the dependent and independent variables were entered into a single exploratory factor analysis. This analysis produced four factors, each of which had an eigenvalue greater than 1.0, and collectively accounted for 62% of the variance in the data. The first factor explained about 30% of the total variance. Since a single factor did not account for most of the variance, these results suggest that common method bias is unlikely to be a significant issue in the collected data. In addition, path coefficients have different levels of significance, and the non-significance of control variables provide further support for the likely absence of common method bias issues.

Second, our results are derived from data associated with users of a single vendor's software development platform. This approach was used to establish controls against variation in the capabilities of the technology platform and to focus on the integration of tacit and explicit knowledge and process formalization. However, this approach limits the generalizability of the results and future researchers should replicate this study across tools and development platforms.

References

Alavi, M. (2000) "Managing Organizational Knowledge" in R. W. Zmud (ed.) *Framing the Domains of IT Management Research: Glimpsing the Future Through the Past*, Cincinnati, OH: Pinnaflex Educational Resources.

Alavi, M. and D. E. Leidner (2001) "Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues", *MIS Quarterly,* (25) 1, pp. 107-136.

Alter, S. (2001) "Which Life Cycle - Work, Information, or Software?" *Communications of the AIS,* (7) 17, pp. 1-54.

Barclay, D., C. Higgins, and R. Thomson (1995) "The Partial Least Squares Approach to Causal Modeling, Personal Computer Adoption and Use as an Illustration", *Technology Studies,* (2) 2, pp. 285-309.

Baroudi, J. J., M. H. Olson, and B. Ives (1986) "An Empirical Study of the Effect of User Involvement on System Usage and Information Satisfaction", *Communications of the ACM,* (29) 3, pp. 232-238.

Baskerville, R. and J. Pries-Heje (2001) "e-Methodology: Towards a Systems Development Methodology for e-Business and e-Commerce Applications", in S. Elliot (ed.) *Developing a Dynamic, Integrative, Multi-disciplinary Research Agenda in e-Commerce/e-Business*, Salzburg, Germany: IFIP and OCG.

Benko, C. and W. McFarlan (2003) *Connecting the Dots: Aligning Projects with Objectives in Unpredictable Times,* Boston, MA: Harvard Business School Press.

Brown, J. and P. Duguid (1998) "Organizing Knowledge", *California Management Review,* (40) 3, pp. 90-111.

Carlile, P. R. (2002) "A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development", *Organization Science,* (13) 4, pp. 442-455.

Chen, S. (2005) "Task Partitioning in New Product Development Teams: A Knowledge and Learning Perspective", *Journal of Engineering and Technology Management,* (22) 4, pp. 291-314.

Chin, W. W., and A. Gopal (1995) "Adoption Intention in GSS: Relative Importance of Beliefs", *Data Base Advances,* (26) 2 & 3, pp. 42-63.

Chin, W. W. (1998a) "The Partial Least Squares Approach to Structural Equation Modeling" in G. Marcoulides (ed.) *Modern Methods for Business Research*, Mahwah, NJ: Lawrence Erlbaum Associates, pp. 295-336.

Chin, W. W. (1988b) "Issues and Opinion on Structural Equation Modeling", *MIS Quarterly,* (22:1), pp. vii - xvi.

Chin, W. W., B. L. Marcolin, and P. R. Newstead (2003) "A Partial Least Squares Latent Variable Modeling Approach for Measuring Interaction Effects: Results from a Monte Carlo Simulation Study and an Electronic-mail Emotion/Adoption Study", *Information Systems Research,* (14) 2, pp. 189-217.

Compeau, D. R., and C. A. Higgins (1995) "Computer Self-efficacy: Development of a Measure and an Initial Test", *MIS Quarterly,* (19) 2, pp. 189-211.

Davenport, T. and J. Short (1990) "The New Industrial Engineering: Information Technology and Business Process Redesign", *Sloan Management Review,* (31) 4, pp. 11-27.

Davenport, T. H., D. W. D. Long, and M. C. Beers (1998) "Successful Knowledge Management Projects", *Sloan Management Review,* (39) 2, pp. 43-57.

Davenport, T. H. and L. Prusak (2000) *Working knowledge: How Organizations Share What They Know*, Boston, MA: Harvard Business School Press

DeFranco-Tommarillo, J and F. P. Deek (2004) "Collaborative Problem Solving and Groupware for Software Development", *Information Systems Management,* (21)1, pp. 67-80.

Doll, W. J and G. Torkzadeh (1989) "A Discrepancy Model of End-User Computing Involvement", *Management Science*, (35) 10, pp. 1151-1171.

Dougherty, D. (1992) "A Practice-Centered Model of Organizational Renewal through Product Innovation", *Strategic Management Journal,* (13) 1, pp. 77-93.

Fichman, R.G. and C. F. Kemerer (1997) "The Assimilation of Software Process Innovations: An Organizational Learning Perspective", *Management Science*, (43)10, pp. 1345 - 1363.

Finlay, P. N. and A. C. Mitchell (1994) "Perceptions of Benefits from Introduction of CASE: An Empirical Study", *MIS Quarterly,* (18) 4, pp. 353-370.

Glen, P. (2006) "Detecting Disaster Projects", *ComputerWorld*, 40(6), pp. 39.

Grant, R. M. (1996) "Toward a Knowledge-Based Theory of the Firm", *Strategic Management Journal,* (17) 2, pp. 109-122.

Hansen, M. T., N. Nohria, and T. Tierney (1999) "What's Your Strategy for Managing Knowledge?" *Harvard Business Review,* (77) 2, pp. 106-117.

Hawryszkiewycz, I.T. (2005) "A Metamodel for Modeling Collaborative Systems", *The Journal of Computer Information Systems*, (45) 3, pp. 63-72.

Hoegl, M., K. P. Parboteeah, and C. L. Munson (2003) "Team-Level Antecedents of Individuals' Knowledge Networks", *Decision Sciences,* (34) 4, pp. 741-771.

Hong, P. Doll, W. J., A. Y. Nahm, and X. Li (2004) "Knowledge Sharing in Integrated Product Development", *European Journal of Innovation Management,* (7) 2, pp. 102-112.

Hoopes, D. G. and S. Postrel (1999) "Shared Knowledge, "Glitches," and Product Development Performance", *Strategic Management Journal,* (20) 9, pp. 837-851.

Howell, J. M., and C. A. Higgins (1990) "Champions of Technological Innovation", *Administrative Science Quarterly,* (35) 2, pp. 317-341.

Iivari, J., R. Hirscheim, and H. K. Klien (2001) "A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches", *Journal of Management Information Systems,* (17) 3, pp. 179-218.

Jarvis, C. B., S. B. Mackenzie, and P. M. Podsakoff (2003) "A Critical Review of Construct Indicators and Measurement Model Misspecification in Marketing and Consumer Research", *Journal of Consumer Research,* (30) 2, pp. 199-218.

Keil, M., P. E. Cule, K. Lyytinen, and R. C. Schmidt (1998) "A Framework for Identfiying Software Project Risks", *Communications of the ACM,* (41) 11, pp. 76-83.

Keil, M., J. Mann, and A. Rai (2000) "Why Software Projects Escalate: An Empirical Analysis of Four Theoretical Models", *MIS Quarterly,* (24) 4, pp. 631-664.

Kirsch, L. J. (1996) "The Management of Complex Tasks in Organizations: Controlling the Systems Development Process", *Organization Science,* (7) 1, pp. 1-21.

Kogut, B. and U. Zander (1996) "What Do Firms Do? Coordination, Identity and Learning", *Organization Science,* (7) 3, pp. 502-518.

Kwan, M. M. and P. Balasubramanian (2003) "KnowledgeScope: Managing knowledge in context", *Decision Support Systems,* (35)4, pp. 467-482.

Löhmoeller, J. B (1984) *LVPS 1.6 Program Manual: Latent Variable Path Analysis with Partial Least Squares Estimation*, Cologne, Germany: Central Archives for Empirical Social Research.

Maier, R. and U. Remus (2003) "Implementing Process-oriented Knowledge Management Strategies", *Journal of Knowledge Management,* (7) 4, 2003 pp. 62-75.

McKeen, J. D., and T. Guimaraes (1997) "Successful Strategies for User Participation in Systems Development", *Journal of Management Information Systems*, (14) 2, pp. 113-150.

Melcher, A., W. Acar, P. Dumont, and M. Khouja (1990) "Standard-Maintaining and Continuous-Improvement Systems: Experiences and Comparisons", *Interfaces,* (20) 3, pp. 24-41.

Mi, P. and W. Scacchi (1992) "Process Integration in CASE Environments", *IEEE Software,* (13) 3, pp. 45-53.

Nidumolu, S. and M. Subramani (2004) "The Matrix of Control: Combining Process and Structure Approaches to Managing Software Development", *Journal of Management Information Systems,* (20) 3, pp. 159 - 196.

Nilkanta, S. and R. W. Scammell (1990) "The Effect of Information Sources and Communication Channels on the Diffusion of an Innovation in a Database Environment", *Management Science,* (36) 1, pp. 24-40.

Nonaka, I. A. (1994) "Dynamic Theory of Organizational Knowledge Creation", *Organization Science,* (5)1, pp. 4-37.

Nonaka, I. and H. Takeuchi (1995) *The Knowledge-Creating Company*, New York, NY: Oxford University Press.

Nonaka, I., and N. Konno (1998) "The Concept of 'Ba': Building a Foundation for Knowledge Creation", *California Management Review,* (40)3, pp. 40-54.

Nonaka, I., and R. Toyama (2003) "The knowledge-creating theory revisited: Knowledge creation as a synthesizing process", *Knowledge Management Research & Practice,* (1)1, pp. 2-10.

Okhuysen, G. A. and K. M. Eisenhardt (2002) "Integrating Knowledge in Groups: How Formal Interventions Enable Flexibility", *Organization Science,* (13) 4, pp. 370-386.

Orlikowski, W. J. (2002) "Knowing in Practice: Enacting a Collective Capability in Distributed Organizing", *Organization Science,* (13) 3, pp. 249-273.

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

Patnayakuni, R., A. Rai, and A. Tiwana (2006) "Software Development Process Improvement: A Knowledge Integration Perspective," *IEEE Transactions on Engineering and Management*, forthcoming.

Paulk, M. C., C. V. Weber, B. Curtis, and M. B. Chrissis (1995) "The Capability Maturity Model: Guidelines for Improving the Software Process", Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Podsakoff, P. M. and D. W. Organ (1986) "Self-reports in Organizational Research: Problems and Prospects", *Journal of Management,* (12) 4, pp. 531-544.

Polanyi, M. (1966) *The Tacit Dimension*, Londong, UK: Routledge and Keoan.

Pratt, M. (2006) "What Do Users Want?" *ComputerWorld*, (40) 26, pp. 40-41.

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

Rai, A. and R. Patnayakuni (1996) "A Structural Equation Model for CASE Adoption Behavior", *Journal of Management Information Systems,* (13) 2, pp. 205-234.

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

Rozeboom, W. W. (1979) "Sensitivity of a Linear Composite Predictor Items to Differential Item Weighting", *Psychometrica,* (44) 3, pp. 289-296.

Sarbaugh-Thompson, M. and M. S. Feldman (1998) "Electronic Mail and Organizational Science: Does Saying "Hi" Really Matter?" *Organization Science,* (9) 6, pp. 685-698.

Schmidt, R., K. Lyytinen, M. Keil, and P. Cule (2001) "Identifying Software Project Risks: An International Delphi Study", *Journal of Management Information Systems,* (17) 4, pp. 5-36.

Sethi, V., & W. King (1991), "Construct Measurement in Information Systems Research: An Illustration in Strategic Systems", *Decision Sciences*, (22) 3, pp. 455-472.

Sharma, S., R. M. Durand, and O. Gur-Arie (1981) "Identification and Analysis of Moderator Variables", *Journal of Marketing Research,* (18) 3, pp. 291-300.

Straub, D.W. (1989) "Validating Instruments in MIS Research"*, MIS Quarterly*, (13) 2, pp. 147-165.

Tiwana, A. and M. Keil (2004) "The One-Minute Risk Assessment Tool", *Communications of the ACM,* (47) 11, pp. 73-77.

Tsoukas, H. (1996) "The Firm as a Distributed Knowledge System: A Constructionist Approach", *Strategic Management Journal,* (17) 1, pp. 11-25.

Tuomi, I. (1999) "Data is more than knowledge: Implications of the reversed hierarchy for knowledge management and organizational memory", *Journal of Management Information Systems,* (16) 3, pp. 103-117.

Vessey, I. and S. Conger (1993) "Learning to Specify Information Requirements: The Relationship Between Application and Methodology", *Journal of Management Information Systems,* (10) 2, pp. 177-201.

Wallace, L., M. Keil, and A. Rai (2004) "Understanding Software Project Risk: A Cluster Analysis", *Information and Management,* (42) 1, pp. 115-125.

Walz, D. B., J. J. Elam, and B. Curtis (1993) "Inside a Software Design Team: Knowledge Acquisition, Sharing and Integration", *Communications of the ACM*, 36(10), pp. 62-77.

Zmud, R. W. (1982) "Diffusion of Modern Software Practices: Influence of Centralization and Formalization", *Management Science,* (28) 10, pp. 1421-1431.

About the Authors

Ravi Patnayakuni is an Associate Professor of information systems in the Department of Economics and Management Information Systems at the University of Alabama in Huntsville. His research focuses on, digital supply chains, IT business value, IT implementation, systems development and knowledge management. Ravi received his D.B.A from Southern Illinois University at Carbondale and has held positions at Temple University and The University of Melbourne, Australia prior to his current appointment. His research has been published in *Journal of Management Information Systems*, *MIS Quarterly*, *Communications of the ACM*, *Omega*, *Communications of the AIS*, and *Information Systems Journal.*

Cynthia P. Ruppel is an Associate Professor of information systems in the Department of Economics and Management Information Systems at the University of Alabama in Huntsville. She received her PhD. in MIS from Kent State University. Her research interests include the use of telecommunications to conduct business such as in telecommuting, ecommerce and supply chains as well as the impacts of innovation adoption and diffusion in organizations. Her work has been published in *IEEE Transactions on Professional Communications, Database for Advances in Information Systems, Information Resource Management Journal, Business Process Management Journal, e-Service Journal,* and other journals.

Arun Rai is Regents' Professor and Harkins Chaired Professor in the Center for Process Innovation and Department of Computer Information Systems at Georgia State University's Robinson College of Business. His research interests include digitally-enabled supply chain management, diffusion and impacts of information technology, and management of systems delivery. His work has been published in *Decision Sciences, European Journal of Operations Research, IEEE Transactions on Engineering Management, Information Systems Research, Journal of Management Information Systems,* and *MIS Quarterly,* among others. He has served, or serves, on the editorial boards for *Decision Sciences*, *IEEE Transactions on Engineering Management, Information Systems Research, MIS Quarterly,* and others. Leading corporations, including A.T. Kearney, Bozell Worldwide, Daimler-Chrysler, Gartner, IBM, Intel, UPS, SAP, among others, have sponsored his research.