

The Effectiveness of Knowledge Transfer Portfolios in Software Process Improvement: A Field Study

Sandra A. Slaughter

David A. Tepper School of Business, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213,
sandras@andrew.cmu.edu

Laurie J. Kirsch

Joseph M. Katz Graduate School of Business, University of Pittsburgh, Pittsburgh, Pennsylvania 15260,
lkirsch@katz.pitt.edu

Because of challenges often experienced when deploying software, many firms have embarked on software process improvement (SPI) initiatives. Critical to the success of these initiatives is the transfer of knowledge across individuals who occupy a range of roles in various organizational units involved in software production. Prior research suggests that a portfolio of different mechanisms, employed frequently, can be required for effective knowledge transfer. However, little research exists that examines under what situations differing portfolios of mechanisms are selected. Further, it is not clear how effective different portfolio designs are. In this study, we conceptualize knowledge transfer portfolios in terms of their *composition* (the types of mechanisms used) and their *intensity* (the frequency with which the mechanisms are utilized). We hypothesize the influence of organizational design decisions on the composition and intensity of knowledge transfer portfolios for SPI. We then posit how the composition and intensity of knowledge transfer portfolios affect performance improvement. Our findings indicate that a more intense portfolio of knowledge transfer mechanisms is used when the source and recipient are proximate, when they are in a hierarchical relationship, or when they work in different units. Further, a source and recipient select direction-based portfolios when they are farther apart, in a hierarchical relationship, or work in different units. In terms of performance, our results reveal that the fit between the composition and intensity of the knowledge transfer portfolio influences the recipient's performance improvement. At lower levels of intensity direction-based portfolios are more effective, while at higher levels of intensity routine-based portfolios yield the highest performance improvement. We discuss the implications of our findings for researchers and for managers who want to promote knowledge transfer to improve software processes in their organizations.

Key words: knowledge transfer mechanism portfolios; portfolio intensity; portfolio composition; knowledge transfer; software process improvement; management of information systems

History: V. Sambamurthy, Senior Editor. This paper was received on February 15, 2005, and was with the authors 7½ months for 1 revision.

Introduction

In recent years, firms have invested heavily in software as information technology (IT) plays a critical role in many aspects of the value chain. However, firms often fail in their attempts to build and deploy software (Gaudin 2003), prompting a growing interest in software process improvement (SPI) initiatives (Herbsleb et al. 1997). These initiatives involve the introduction of specific practices and measures designed to improve the deployment and subsequent maintenance of information systems (IS) (Grady 1997).

Process improvement of any kind—including SPI—is not simply a matter of individuals embracing incremental changes. Rather, armed with new knowledge, individuals in various roles and units fundamentally rethink work patterns and relationships, developing new cognitive frameworks and schemas (Mitki et al. 1997, Spencer 1994) and embed these new structures into their work practices (Ravichandran and Rai 2003). SPI transcends discrete projects and systems, and comes from recognizing patterns *across* projects and systems to design and institutionalize new work practices (Mathiassen and Pourkomeylian

2003). Improvement emanates from a deep and broad understanding of current processes and practices, their patterns, and implications. Developing this kind of understanding requires *knowledge transfer* about software processes and practices between individuals occupying various organizational roles and located in different work units.

Knowledge transfer is a dyadic exchange in which a recipient learns and applies knowledge transmitted from a source (Ko et al. 2005, Argote and Ingram 2000). Knowledge is transferred via mechanisms such as technology, personnel movement, and blueprints (Darr et al. 1995, Argote 1993, Epple et al. 1991, Adler 1990). Recently, researchers have argued that effective knowledge transfer is dependent, in part, on repeatedly using combinations of mechanisms rather than relying on the single use of a single mechanism (Rulke et al. 2000, Malhotra et al. 2001, Argote 1999). Extending these arguments, we define a *portfolio of knowledge transfer mechanisms* as the set of mechanisms used by a source to transfer knowledge to a recipient. Further, we characterize portfolios by their *composition* (the types of mechanisms used) and their *intensity* (the frequency with which the mechanisms are utilized).

The purpose of our study is to examine the role that knowledge transfer portfolios play in SPI. In particular, we examine factors that influence the structure of the portfolio in terms of its composition and intensity, as well as the effectiveness of various portfolios. Our study is situated in the context of an SPI initiative undertaken by three distinct units—Development, Maintenance, and Quality Assurance (QA)—within a large IS department in one firm.

To the best of our knowledge, our study is one of the first to conceptualize and empirically investigate knowledge transfer portfolios. We elaborate theoretically grounded dimensions of knowledge transfer portfolios—composition and intensity. Our focus on portfolios draws attention to the range of mechanisms used, and the way in which the mechanisms are combined, to transfer knowledge about software processes across roles and units. Our study also provides insight into the effectiveness of various knowledge transfer portfolio designs by examining the fit between portfolio composition and intensity and performance improvement.

We next review the relevant literatures on SPI and on knowledge transfer, and develop our model and hypotheses. We then describe the methodology, present the results of our analysis, and discuss their implications. We conclude by identifying the contributions and limitations of our study and suggest areas for future research.

Relevant Literature

Software Process Improvement

Improving software development capabilities requires knowledge creation (the accumulation of knowledge) and knowledge embedding, absorbing individually and collectively held knowledge into organizational processes (Ravichandran and Rai 2003). While the knowledge required for process improvement may include some simpler or more explicit aspects, to effectively embed and apply the knowledge requires a more complex understanding of the context and task (Mathiassen and Pourkomeylian 2003). For example, consider an IS manager who wants to adapt development processes to facilitate the maintenance of systems. To enable this SPI, the manager can assign a maintainer to a development team to transfer knowledge about which coding practices yield code that is difficult or costly to maintain. Knowledge is transferred to the developer from the maintainer over the time period of the SPI initiative as the developer regularly interacts with and observes the maintainer. In due course, this knowledge transfer may yield new processes like peer design reviews that improve maintenance efficiency by identifying problematic designs early in the software life cycle.

Improving the practice of software development and maintenance therefore requires the transfer of knowledge between source and recipient dyads who are located within and across organizational units (Mitki et al. 1997). Moreover, it requires continuous interaction between source and recipient over time; multiple transfers utilizing multiple mechanisms may be required so that a recipient understands the value of the innovations and begins to incorporate them into specific work practices (Ravichandran and Rai 2003). Institutionalizing these new work practices, however, takes time, suggesting that individual performance improvement may not be observed after a

single knowledge transfer event. Thus, in our study, the portfolio of mechanisms used by a source to transfer software process knowledge to a recipient represents the *accumulation* of single transfers between a particular source and recipient over the timeframe of the process improvement period. In our example above, the portfolio of mechanisms includes team membership, observation, and source-recipient interactions. In this portfolio, observation and interaction are utilized multiple times to transfer knowledge about coding practices, representing repeated transfers of this knowledge as the SPI initiative unfolds. As illustrated in our example, knowledge transfer for the purpose of SPI is not restricted to a particular part of the IS department, but it occurs across units and roles.

Knowledge Transfer Mechanisms and Portfolios

Knowledge is information possessed by an individual that, when combined with other personal dimensions such as experience and reflection, becomes a basis for action (Grover and Davenport 2001). For example, information about software defect patterns in a particular system becomes “knowledge” when it is combined with an understanding of system development and business processes to improve the design of future systems. Just as knowledge about defect patterns can become a basis for action, so too can knowledge of other software product and process attributes (e.g., software size, productivity, or effort) spur changes in development and maintenance practices. Moreover, transferring knowledge among key stakeholders means that software development processes can be shaped for improved efficiency and effectiveness above and beyond what any one individual can accomplish (Ravichandran and Rai 2003, 2000).

Generally speaking, knowledge is transferred “by moving people, technology, or structure to the recipient organization or by modifying the people (e.g., through training), technology, and structure of the recipient organization” (Argote 1999, p. 145). Specific transfer mechanisms include training, observation of experts, tooling, assembly line layouts, routines, and meetings (Contu and Willmott 2003, Thomas-Hunt et al. 2003, Birkinshaw et al. 2002, Adler 1990).

Though much of the empirical literature seems to assume that knowledge transfer is composed of a *single* movement of knowledge from a source to a recipient utilizing a *single* mechanism, some researchers have begun to take a broader view (McGrath and Argote 2001). Hansen (1999), for example, found that repeated interactions between individuals with strong ties facilitated knowledge acquisition. The results of Carlile’s (2004) study suggest that successful knowledge transfer requires multiple iterations of sharing through which individuals develop common language, understanding, and interests. In an experiment examining knowledge sharing, Paulus and Yang (2000) found that more frequent interactions among group members engaged in idea generation resulted in better performance than when ideas were generated by individuals but not shared with group members. These studies highlight the importance of the *intensity* of the portfolio; that is, the frequency with which the mechanisms are used.

Not only does the literature suggest that a portfolio varies in intensity, but there is also evidence to suggest that its *composition* (i.e., the types of mechanisms included) matters. Argote and Ingram (2000) note that though transferring knowledge by moving members is often difficult, transferring knowledge with a combination of task and tool elements can be effective. Others have observed similar findings about the use of multiple kinds of mechanisms. Galbraith (1990), for example, found that effective knowledge transfers utilize a combination of people and technology mechanisms. In their study, Rulke et al. (2000) observed a positive relationship between the use of multiple mechanisms of knowledge transfer (including training, newsletters, observation, personal contacts, and conference presentations) and learning outcomes. This research establishes the importance of using multiple types of mechanisms, but it raises the question of how to meaningfully categorize these mechanisms to better understand their use and impact. We turn to this question in the next section.

A Typology of Knowledge Transfer Mechanisms

In a conceptual paper, Grant (1996) argues that the primary role of the firm is knowledge integration, or the process of combining the specialized knowledge

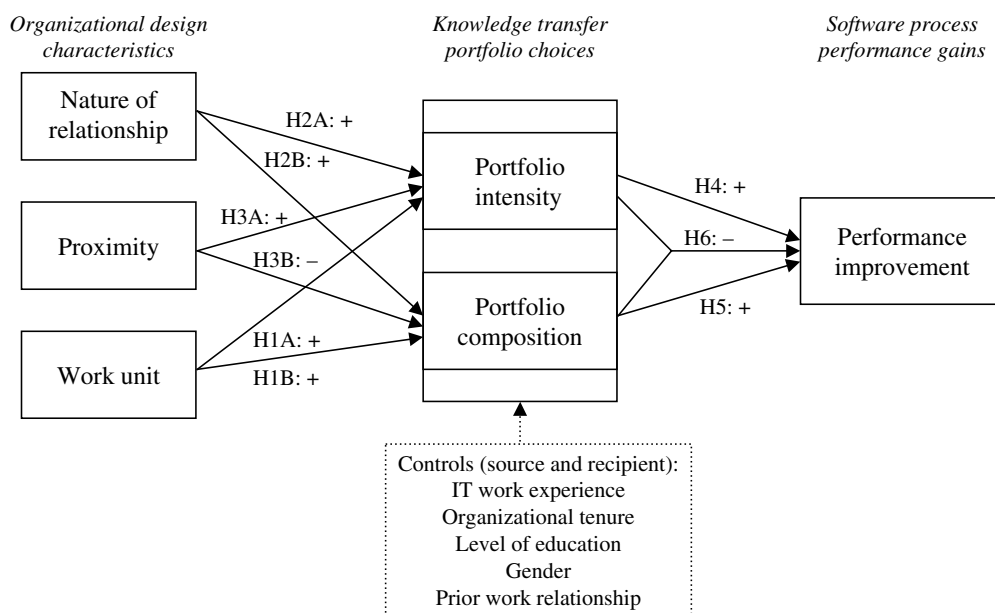
of a number of individuals. His concept of integration includes the transfer and replication of knowledge from individual to individual in an organization. According to Grant (1996), there are two fundamental categories of mechanisms that can be used to integrate knowledge: directions and organizational routines. Mechanisms classified as “directions” are explicit rules and instructions, such as standard operating procedures or plans. In contrast, organizational routines are those mechanisms that allow for integrating knowledge without explicitly communicating it: “...the essence of an organizational routine is that individuals develop sequential patterns of interaction which permit the integration of their specialized knowledge without the need for communicating that knowledge” (Grant 1996, p. 379). Surgical teams and project teams use routines to transfer knowledge when individuals embrace a common understanding of roles and interactions, bring their own expertise to bear on the task, and work closely together to accomplish their objective. As explained below, Grant’s (1996) typology is well suited to the context of our study.

Directions transfer explicit knowledge relating to SPI by coding the knowledge into rules and procedures. The software process improvement context,

with its emphasis on documentation, standards, metrics, and training, provides ample opportunities to make use of such mechanisms to transfer knowledge across roles and units in the IS department. On the other hand, organizational routines are more implicit means of transferring knowledge, which emphasize the movement of knowledge “without language” (Nonaka 1994, p. 19). Organizational routines can be found in SPI in the form of observing coding experts in their work, imitating the use of best design practices, and informal interactions between a system developer and a maintainer or a quality specialist.

The theoretical foundations, as well as the parsimonious nature, of the directions—organizational routines typology make it attractive for classifying knowledge transfer mechanisms in portfolios. Moreover, given the fit of this typology with the SPI context (as also perceived by Ravichandran and Rai 2003), we felt it was most suited to our objectives and adopted it to describe the composition of knowledge transfer portfolios in our study. We next develop a model (summarized in Figure 1) that examines the antecedents of the intensity and composition of a portfolio of knowledge transfer mechanisms, as well as the consequences for software process performance.

Figure 1 Research Model



Research Model and Hypotheses

As shown in Figure 1, we posit that organizational design characteristics will influence the nature of the knowledge transfer portfolios selected by a source and recipient, and portfolio choices will, in turn, influence the recipient's performance gains from software process improvement. In the context of SPI initiatives, where learning and applying new knowledge are essential, organizational design parameters have important implications for knowledge transfer (Ravichandran and Rai 2003, 2000). Indeed, the structural configuration of an organization has been identified as a critical factor that can facilitate or inhibit process improvement (Mitki et al. 1997) because knowledge tends to move horizontally and vertically along structural lines (Schulz 2001). Thus we examine how portfolio intensity and composition may differ for flows of knowledge across a source and recipient who reside in similar or different work units, and for flows of knowledge between a source and recipient who are in a hierarchical or nonhierarchical relationship. We then consider how the proximity of the source and recipient affects portfolio choices. To shed light on the effectiveness of portfolio intensity and composition, we also examine performance outcomes associated with various portfolio designs.

Work Units and Portfolio Choices

To improve software processes requires understanding the cause and effect relationships between process parameters and outcomes (Ravichandran and Rai 2003). However, the individuals who have knowledge about development processes could be in different work units than the individuals who have knowledge about process outcomes. For example, a maintenance specialist could have detailed knowledge about the performance (such as the response time or error rate) of a particular IS over a period of time. A development specialist could have detailed knowledge about the processes used to create that system, and a QA specialist could have knowledge about how to identify the root causes of quality problems. To understand how the performance problems in this system relate to the development processes used to create it requires the development, maintenance, and QA specialists to transfer their specialized knowledge about process attributes and outcomes.

It is often the case that the greater the diversity of knowledge among the specialists involved, the lower is the level of their shared knowledge, in part because one specialist generally does not develop deep familiarity in another's area of expertise. Prior research suggests that unless the specialists have worked closely together on prior activities, developed expertise or fluency in the language of each other's specialty, or have worked in the same product line or discipline, it is unlikely they will develop shared knowledge (Malhotra et al. 2001, Nelson and Coopriider 1996). This lack of common knowledge makes it difficult for the different specialists to communicate and reduces the efficiency of their knowledge transfer (Carlile 2004, Grant 1996, Demsetz 1991). It is possible that a source and recipient in different work units who continually interact may develop a shared understanding that facilitates their knowledge transfer. However, even with repeated prior interaction, it is not likely that one specialist will develop a complete understanding of another specialist's domain. Thus, a source and recipient who reside in different work units with distinct kinds of expertise will generally need to try harder than a source and recipient in the same work unit to transfer their knowledge. To improve software processes, development, maintenance, and quality specialists residing in different work units *must* share their knowledge. This need to transfer knowledge across units, coupled with the specialized knowledge of the individuals in these units, implies that interunit dyads will repeatedly and frequently transfer knowledge, increasing the intensity of their portfolio.

HYPOTHESIS 1A (H1A). *The intensity of the knowledge transfer portfolio selected will be higher when the source and recipient work in different units than when they work in the same unit.*

We further expect that specialists who are in different work units will be more likely to use directions than routines to transfer knowledge. This is because routines can only be leveraged when there are repeated and direct interactions between the source and recipient who share a common frame of reference. As noted by Nonaka (1994, p. 19), "without some form of shared experience, it is extremely

difficult for people to share each other's thinking processes. The mere transfer of information will often make little sense if it is abstracted from nuanced emotions and nuanced contexts that are associated with shared experiences." Software developers have a shared understanding of the process and context in which user requirements and technology are transformed into functional information systems. Thus, an improvement to a software testing procedure proposed by one developer does not require extensive directions to be comprehended and used by another developer. Instead, this knowledge can be transferred via organizational routines (e.g., the source could demonstrate the new procedure while working with the recipient on a project, or the recipient could observe the source using the new procedure). In contrast, even if they have interacted frequently in the past, a software developer and a QA specialist could still lack a complete shared understanding of the nuances of the development context (because the developer focuses on development activities while the quality specialist focuses on quality-related activities). Thus the developer must be more explicit in documenting and codifying the new procedure for the QA specialist (than he would be with another developer) to understand its implications for software quality. This implies:

HYPOTHESIS 1B (H1B). The use of directions in the knowledge transfer portfolio selected will be higher when the source and recipient work in different units than when they work in the same unit.

Hierarchical Relationships and Portfolio Choices

An important aspect of the source-recipient relationship is whether it is hierarchical or nonhierarchical. Because organizational structures often dictate the paths of communication, there typically are frequent information exchanges along the hierarchy (Hinds and Kiesler 1995). In knowledge transfer, substantial knowledge flows have been observed between subordinates and superiors, especially when the knowledge is new and unsettled (Schulz 2001).

A great deal of the knowledge about software process diagnosis and interpretations flows between a project manager and a software developer or maintainer, or between senior management and subordinates, rather than between peers. This is because the

formal hierarchy provides channels through which knowledge can easily flow. For example, if a project manager is interested in improving the development process, she examines defects in current projects, identifies trends in the defects, and determines whether patterns in defects can be traced to specific development processes. Knowledge about defect patterns, which becomes one basis for improving development processes, is transferred from the manager to a developer using existing channels (mechanisms), such as regularly scheduled meetings and formal memos. It is important to note that either person in the hierarchical relationship could be the source of the knowledge transfer; that is, the source is not always the superior. The project manager, in the above example, could transfer the knowledge about defect patterns to her superior as well as to her subordinate. Frequent interactions and knowledge flow along the hierarchy, as compared to peer-to-peer knowledge transfer, seems to accelerate the identification and successful use of best practices (Ravichandran and Rai 2000). Thus we expect:

HYPOTHESIS 2A (H2A). The intensity of the knowledge transfer portfolio selected will be higher when the source and recipient are in a hierarchical relationship than when they are not.

With respect to portfolio composition, we anticipate that peers would rely more on routines while superiors and subordinates would rely more on directions. One reason is efficiency. As Grant (1996) explains, the efficiency of knowledge transfer requires economizing on the amount of communication required for transfer to occur. Organizational structures and, in particular, bureaucracies, are very efficient in transferring knowledge along the hierarchy because directions can be used as the predominant integrating mechanism, achieving higher levels of coordination with less communication. For example, a senior manager could develop a new policy for software testing. Codifying the policy into a set of explicit rules and instructions is an economical way to transfer knowledge about the new policy to subordinates.

Directions leverage the detailed articulation of instructions, while routines typically rely upon a much more limited set of cues and responses that serve not so much as to communicate knowledge, but rather

to permit a sequencing of an individual's application of knowledge inputs (Grant 1996). As we reasoned earlier, when a source and recipient share a common understanding, they can more easily rely on routines to transfer knowledge. We expect that peers share a more common frame of reference than superiors and subordinates, and thus are more likely to use routines than a superior and subordinate:

HYPOTHESIS 2B (H2B). The use of directions in the knowledge transfer portfolio selected will be higher when the source and recipient are in a hierarchical relationship than when they are not.

Proximity and Portfolio Choices

Proximity is an important organizational design parameter. The literature suggests that collocating individuals is vital to developing shared goals, deep understanding, and information sharing, particularly when individuals are in different disciplines or when the task is uncertain or complex (e.g., Argote 1999, Crowston and Kammerer 1998). In software development, Teasley et al. (2002, p. 671) have documented the beneficial effects of radical collocation or "war rooms." Proximity has been shown to increase opportunities for interactions between individuals (Kraut and Streeter 1995). In addition, the literature suggests that the beneficial effects of collocation fade very quickly, at distances beyond 30 meters (Allen 1977). Thus, even relatively short distances between individuals can significantly affect their communication and information-sharing behaviors. Indeed, despite the advantages of new collaboration technologies, experts concur that "distance still matters" (Olson and Olson 2000, p. 139).

Because proximity fosters personal contact and communication, it seems likely that a colocated source and recipient will transfer knowledge more intensely than a source and recipient who are more distant: Individuals who are physically proximate have more opportunity to use personal mechanisms of transfer (Argote 1999), in addition to technology and structural mechanisms that represent the most likely choices for a source and recipient who are separated in space. All else being equal, then, it seems likely that a proximate source-recipient dyad will transfer knowledge more frequently. For example, a QA specialist who is located close to developers could

leverage knowledge transfer mechanisms that rely on personal interactions more frequently and readily than a quality specialist who is located in another part of the building or another city. This suggests:

HYPOTHESIS 3A (H3A). The intensity of the knowledge transfer portfolio selected will be higher the closer the source and recipient.

We also expect that proximity influences the composition of the knowledge transfer portfolios selected. A proximate source and recipient can leverage informal procedures and implicit communications of routines to transfer knowledge, while a distant source and recipient may be restricted to codified directions for easy transport to remote locations. For example, a QA specialist who is located close to developers and maintainers could leverage routines such as observation of experts, face-to-face meetings, and informal activities, while the more distant quality specialist may have to use directions such as computer-based reports and repositories that can be accessed across large distances. Thus we posit:

HYPOTHESIS 3B (H3B). The use of directions in the knowledge transfer portfolio selected will be lower the closer the source and recipient.

Portfolio Choices and Performance Improvement

The characteristics of the knowledge transfer portfolio used—its composition and intensity—are expected to impact the performance of the recipient. The portfolio represents the set of mechanisms used by a source to transfer knowledge to a recipient over the life of the SPI initiative. A successful knowledge transfer event implies that the recipient has applied the knowledge supplied by the source. Realizing performance gains requires that the transferred knowledge be embedded in new work practices, which are sustained over time to become institutionalized (Ravichandran and Rai 2003). It is the institutionalization of these work practices—their routinization and regular use—that drives improvement in performance. As the recipient receives an increasing amount of knowledge from a variety of sources over the life of the process improvement period, he is learning techniques and approaches to enhance his own software development practices, and therefore his performance should improve as the SPI initiative goes forward.

However, successful knowledge transfer, in the sense that a recipient applies knowledge from a source, does not guarantee that new practices are institutionalized and performance gains are realized. Generating positive impacts on performance in the long term would seem to depend on whether two conditions are met: (1) that the correct knowledge is transferred by the source and (2) that the recipient correctly applies the knowledge (Mitki et al. 1997). These ideas are explored next as we develop specific hypotheses that relate portfolio composition and intensity to performance improvement.

We expect that the intensity of the portfolio of knowledge transfer mechanisms will have a positive impact on performance improvement. An intense portfolio suggests that the source is repeatedly and frequently transferring knowledge to a recipient. The source is not relying on a single knowledge transfer event, but instead transferring the knowledge to the recipient on multiple occasions. When knowledge is rich and contextualized, such as the knowledge required to improve and institutionalize work processes, repeated transfers may be necessary to generate a deep understanding of the process on the part of the recipient and to enable the recipient to embed the new process into ongoing work practices (Ravichandran and Rai 2003, Alavi 2000). A single transfer of knowledge may not be as effective as repeated transfers (Rulke et al. 2000). For example, consider a scenario where a quality specialist is transferring knowledge about how to use root cause analysis to determine the source of software defects. The initial transfer could involve a lecture, and the recipient may be able to apply the concept after that lecture. However, it is likely that if the lecture is followed by additional lectures, demonstrations, and tutorials, the recipient may be better able to understand and more effectively integrate root cause analysis into software design practices. Thus, in the SPI context, it seems likely that more knowledge transfers will result in a recipient who truly understands the knowledge being transferred, increasing the likelihood that the recipient will be able to apply that knowledge effectively, and thus generate positive performance gains. This argument is consistent with prior research that has found a positive association between frequent knowledge transfer or sharing and performance (Paulus and Yang 2000). This leads us to predict:

HYPOTHESIS 4 (H4). *The higher the intensity of the knowledge transfer portfolio selected by the source, the greater will be the performance improvement of the recipient.*

We also expect portfolio composition to impact performance. Directions represent knowledge that can be codified and articulated, and would seem to facilitate the source's ability to transfer the correct knowledge. By their very nature, directions are less ambiguous and more precise than routines, which lack such explicit specification. When knowledge is transferred via directions (such as standard operating procedures, manuals, and policy statements), it seems reasonable to expect that the recipient will understand the knowledge in a relatively precise manner, and thus be able to apply it correctly. For example, knowledge about how to count function points can be transferred in a manual, and can be applied directly by recipients. In contrast, understanding knowledge that is communicated via routines requires common understanding and shared experiences between the source and recipient. Moreover, it implies that recipients need to interpret and contextualize the knowledge to apply it, suggesting that performance gains may be realized more slowly. Thus, on average, we expect to see that portfolios composed of a greater proportion of directions will yield more performance gains.

HYPOTHESIS 5 (H5). *The greater the use of directions in the knowledge transfer portfolio by the source, the greater will be the performance improvement of the recipient.*

We also posit that composition and intensity interact to affect performance. As Grant (1996) argues, directions are an efficient means of transferring knowledge. This suggests that if a portfolio is largely composed of directions, there is no need to transfer this knowledge repeatedly (that is, intensity should be low). For example, a procedures manual (a direction) does not need to be transferred multiple times over the life of the SPI initiative; once should suffice (low intensity). On the other hand, if routines are used to transfer knowledge, the knowledge is tacit, and thus more amorphous. Successfully transferring this type of knowledge would seem to require multiple transfers (i.e., intensity should be high), such as repeated observation of experts or closely working with specialists. Consistent with this argument, Alavi has noted that "... tacit knowledge is best transferred

through collaboration, shared experience, and rich interpersonal interactions over time (e.g., a mentoring relationship)” (2000, p. 20). Thus we posit that the fit between composition and intensity of the knowledge transfer portfolio will together influence the recipient’s performance improvement, such that high intensity and low (i.e., routine-based portfolio) composition, or low intensity and high (i.e., a direction-based portfolio) composition will yield performance improvement.

HYPOTHESIS 6 (H6). The intensity of the knowledge transfer portfolio interacts with the composition of the knowledge transfer portfolio to affect the performance improvement of the recipient, such that at a lower intensity a portfolio with a greater use of directions will be more effective, while at a higher intensity a portfolio with a greater use of routines will be more effective.

Methodology

We empirically evaluated our conceptual model in a detailed field study of an SPI initiative in an IS department within a large firm. The IS department supports all centralized computing activities for the firm and is located at headquarters. It is organized using the “L” organizational design (Swanson and Beath 1990) in three distinct work units: Development, Maintenance, and QA. The work units in an “L” organizational design are distinct but interdependent. Development designs and implements new IS and major enhancements to existing systems supported by Maintenance, while Maintenance maintains systems that Development has built, makes minor enhancements to them, and supports the users of the systems. QA assesses the quality of IS that are developed and maintained, implements software quality initiatives, and publishes and disseminates software quality information to Development and Maintenance.

Although the work units are interdependent, they do not have significant overlapping knowledge. Indeed, separating Development, Maintenance, and QA into different work units could erect knowledge barriers (for example, the units use different project management systems, metrics, and software engineering tools). At the same time, the interdependencies between the work units motivate the need for knowledge transfer across units to facilitate software process

improvement. For example, to improve the quality of a software design process may require the transfer of knowledge about design processes from Development to Maintenance, and the transfer of knowledge about system quality from Maintenance to Development.

Three distinct roles in the work units are salient to the SPI effort: senior management, project management, and staff. Senior managers establish the overall strategic goals of the IS department as well as the tactical goals of their respective units. Project managers are responsible for day-to-day management of the software projects in their respective units. Members of the technical staff perform the requirements gathering, programming, testing, and implementation tasks for the software projects. Managers in the IS department were interested in our study because they wanted to improve software development and maintenance processes. They viewed the use of quality improvement techniques, software metrics, and best practices in development and maintenance as essential for improved performance. Although a vision for SPI had been articulated, the managers were concerned about whether this initiative would have its intended impact on performance throughout the department.

Data Collection and Coding

Data were collected over one year, beginning when the senior managers had articulated their quality visions, and the SPI initiative commenced. During this time period, we tracked knowledge transfers via observation, interviews, meetings, mail, documents, and manuals. Our data analytic strategy involves a mixed-methods approach, with a quantitative assessment of the qualitative data. To support this strategy, we converted the qualitative information obtained from the various data sources into numerical codes and numbers that could be statistically analyzed; Tashakkori and Teddlie (1998) refer to this data transformation process as “quantitizing.” The following sections describe our data collection and quantitizing process.

Semistructured interviews were conducted in the three work units within the IS department. In particular, all senior managers, project managers, and QA staff were interviewed. In addition, we randomly selected 12 staff members (6 in Development and

6 in Maintenance) to interview. Before conducting the interviews, protocols were developed that included questions about the SPI initiative. Interviews were either tape-recorded or copious notes were taken during the interview and were transcribed immediately afterward. The observation process revealed additional behaviors and environmental conditions (such as the posting of SPI data on bulletin boards) related to knowledge transfer. The first author, as a passive observer, attended a number of staff meetings about SPI, software metrics counting meetings, and software quality training sessions during the data collection period. Detailed and extensive notes were taken during these meetings and sessions, and they were transcribed immediately afterward. The first author also kept a daily diary to record general observations about the organizational environment.

To supplement and validate the interview and observational data, relevant archival data (documents) that spanned the time period of the SPI initiative were collected. Examples include a quality metrics white paper that defines particular metrics, identifies who uses them, and describes how they are used, and memos about the SPI initiative that describe the implementation of various improvements (such as training on quality or the use of new templates). We created an annotated electronic bibliography in chronological order to briefly identify and describe each archival document.

Identifying Knowledge Transfers. In the first round of coding, we examined the interview, observation, and archival data, and collaboratively identified each knowledge transfer relating to SPI. *In identifying knowledge transfers, we did not include regular communications between a source and recipient. Rather, following Argote (1999), we ensured that recipients applied the knowledge.* Consider an example shown in the top half of Figure 2 where a recipient receives information about defect patterns, and applies the knowledge to develop a test plan template. Transmitting information about defects does not by itself constitute a knowledge transfer, but understanding the cause and effect relationship between defects and development processes and applying that knowledge to devise a test plan template does.

Figure 2 A Coded Knowledge Transfer

“My team [Source: *Development, Staff X, Y, and Z*] reports and analyzes development defects every month. We use a Tsuda matrix [Mechanism (Directions): *Tsuda Matrix*] to determine whether there are any patterns in the defects that can be traced to development processes. For example, last month my team found that test plans were inconsistent and often not written. Based on this, I [Recipient: *Development, Project Manager M*] developed a test plan template.”

Interview Transcript, {name redacted}, Development,
Project Manager

CODING			
Observation	Dyad	Portfolio intensity	Portfolio composition
1	XM	1	1.00 (directions)
2	YM	1	1.00 (directions)
3	ZM	1	1.00 (directions)

Categorizing Knowledge Transfer Mechanisms.

In the second round of coding, a research assistant who had not participated in the data collection identified the particular mechanisms used in each knowledge transfer. The mechanisms include a broad range such as training, observation of experts, transferring of personnel, meetings, reports, standards, and IS. In all, 69 unique mechanisms were used in the knowledge transfers. A representative sample of those mechanisms is shown in Table 1.

We initially categorized each of the mechanisms as either “directions” or “routines,” following Grant (1996) and Nonaka and Konno (1998). Specifically, a mechanism was coded as a direction if it represented the codification of tacit knowledge into explicit rules and instructions that provide guidance. Examples include written rules, formal procedures, formulae, IS, policies, guidelines, templates, checklists, formal training, plans, and directives. A mechanism was coded as a routine if it represented an implicit transfer of knowledge without explicit communication such as through interaction or actual performance of a task. Examples include an actual activity or process, commonly understood roles, a demonstration, a meeting, or informal procedures. We then asked an independent coder to also categorize the knowledge transfer mechanisms to provide an independent assessment of our coding. The coder was given a list of the mechanisms and the coding instructions and coded the mechanisms. The coder

Table 1 Mechanisms for Knowledge Transfer

Mechanism	Category	How used
Defect reporting form	Direction	Used to document and describe defects that developers observe in their own or others' software development processes and to suggest causes and solutions.
Development procedures meeting	Routine	An ongoing meeting in which attendees review and discuss defects reported by developers and brainstorm solutions to improve development processes.
MIS dashboard display	Direction	A bulletin board next to the senior managers' offices that displays reports depicting the quality of systems and processes (for example, a graph on defect trends).
Quality training class	Direction	A class held by QA to train developers and maintainers on total quality management concepts and how the concepts apply to software development and maintenance.
Service-level IS	Direction	A set of database queries that periodically calculates the uptime for all applications in operation and reports on trends in the percentage of uptime for different applications. Used to identify problematic applications and to trace the impact of process improvements.
System turnover standards	Direction	A set of metrics against which important characteristics of a new system created by Development are evaluated before the system can be transferred to Maintenance.
Transfer of personnel	Routine	Experienced personnel are transferred and use their experience to help develop better quality applications.
Informal demo of function point counting	Routine	Developers observe other developers using function points to count their projects.

agreed with our categorizations of the mechanisms in all instances but one. The disagreement over the coding of that mechanism was resolved by going back to the transcript of an interview in which the mechanism was described to better understand its context of use. Based on the description in the transcript, the mechanism was categorized according to our original categorization. Table 1 shows the category to which each of the representative mechanisms was assigned.

Identifying the Source and Recipient. A research assistant identified the source and recipient for each

knowledge transfer using organization charts and seating charts to distinguish personnel. A unique code was assigned to each individual in the IS department, of which there were 73 at the time of our study (the CIO, the Software Development Director, the Software Maintenance Director, the QA Director, 3 Project Managers in Development and 3 in Maintenance, 32 Staff in Development, 29 Staff in Maintenance, and 2 Staff in QA). Any of these individuals could assume the "source" or "recipient" role in a knowledge transfer. Based on the organization chart, the research assistant identified the individual's work unit (Development, Maintenance, or QA) and role (Senior Manager, Project Manager, Staff). In a few situations, an individual described a knowledge transfer between other individuals in the IS department or the recipient of the knowledge transfer was not immediately obvious. In these situations, the first author aided the research assistant in identifying the individuals involved.

Source-Recipient Dyad as the Unit of Analysis.

The unit of analysis in this study is the source-recipient *dyad*, i.e., a unique source-recipient pair. For example, information about dyad *AB* [source *A* and recipient *B*] is an observation, information about dyad *BA* [source *B* and recipient *A*] is an observation, information about dyad *AC* [source *A* and recipient *C*] is an observation, and so on. Note that *AB* and *BA* are different dyads because the sources and recipients are different in each dyad.

Measures of each dyad's *portfolio* characterize dimensions of the accumulated *set* of mechanisms used for knowledge transfer between the respective source and recipient over the time period of the SPI initiative. Alavi (2000) argues that contextualized knowledge takes time to transfer using rich mechanisms such as collaboration and mentoring; prior empirical studies have found that multiple mechanisms are often needed to transfer such knowledge (Carlile 2004, Rulke et al. 2000). Therefore, to examine the way in which a source uses a group of mechanisms in a portfolio to transfer knowledge to a recipient, we focus our analysis on the accumulated set or portfolio of knowledge transfer mechanisms. Portfolios included for analysis are those that result in a recipient's application of knowledge from a source. This application might occur after one exchange between source and recipient, or after many such exchanges.

For example, if source *A* uses three mechanisms (two directions and one routine) to transfer the knowledge and recipient *B* applies the knowledge, dyad *AB*'s portfolio has an intensity of three and a composition of 67% directions.

Consider the knowledge transfer shown in Figure 2, where the source is Development Staff. For the sake of illustration, assume there are three development staff involved in this knowledge transfer (with unique identifiers of *X*, *Y*, and *Z*) and one project manager (with a unique identifier of *M*). We would code this as three dyads: *XM*, *YM*, and *ZM*. The knowledge transfer mechanism (Tsuda matrix) would be coded as *Directions* as it represents the codification of tacit knowledge about the meaning and significance of defects into explicit rules and instructions to provide guidance on testing. If this knowledge transfer is the *only* transfer between these dyads, the portfolio intensity would be one for each dyad, and the composition would be 100% directions.

Construct Measurement

Table 2 summarizes the measures of the constructs in the model. The measures of the control variables are straightforward, and are summarized in this table. We describe in more detail the measures of our key constructs in the model: performance improvement, portfolio intensity and composition, and the organizational design characteristics. We measured the SPI outcome of knowledge transfer mechanism portfolio choices—*Performance improvement*—by asking the immediate supervisor of each recipient to rate the extent to which the individual improved the quality, productivity, and effectiveness of his or her work, because of that individual's adoption of SPI (i.e., Project Managers evaluated their respective Staff, Senior Managers evaluated the Project Managers in their respective work units, and the Chief Information Officer evaluated the three Senior Managers). Because SPI requires the institutionalization of new work practices (Ravichandran and Rai 2003), it is unlikely that enhanced performance because of process improvement would be observable early in the SPI initiative but would most likely surface over time. Thus the performance ratings were completed at the end of our study, one year after the commencement of the SPI initiative. The specific questions asked

were: "To what extent do you agree with the following statements: Because {subordinate} uses software process improvements: (1) {subordinate} has improved the quality of his or her work, (2) {subordinate} is more productive in his or her work, (3) {subordinate} is more effective in his or her work." For each question, a 7-point scale was provided that ranged from 1 = strongly disagree to 7 = strongly agree. Having the immediate supervisor evaluate the effectiveness of subordinates strengthens the internal validity of the study because it avoids the self-report bias that can affect performance ratings (Crompton and Wagner 1994). The quality, productivity, and effectiveness measures were highly correlated

Table 2 Construct Measurement

Construct	Measure
Performance improvement	Performance rating by immediate supervisor of the recipient: "To what extent do you agree with the following statements: Because (subordinate) uses software process improvements: (1) (subordinate) has improved the quality of his or her work, (2) (subordinate) is more productive in his or her work, (3) (subordinate) is more effective in his or her work." For each question, a 7-point scale was provided that ranged from 1 = strongly disagree to 7 = strongly agree.
Portfolio intensity	The number of knowledge transfers from a particular source to a particular recipient.
Portfolio composition	The ratio of the number of directions in a source and recipient's portfolio divided by the total number of knowledge transfers in their portfolio.
Proximity	Number of cubicles between a source and recipient. Note that cubicles are all the same size with the exception of managers', which are twice as big.
Nature of relationship	Dichotomous: 1 = source and recipient are in a hierarchical (superior-subordinate or subordinate-superior) relationship; 0 = source and recipient are peers.
Work unit	Dichotomous: 1 = source and recipient are in different work units; 0 = source and recipient are in the same work unit.
IS work experience	Number of years of IS work experience of the source or recipient.
Organizational tenure	Number of years that the source or recipient has worked for the firm.
Education	Highest level of education completed by the source or recipient: 1 = high school; 2 = associate's; 3 = bachelor's; 4 = master's; 5 = Ph.D.
Gender	Dichotomous: 1 = female, 0 = male. Refers to the source's or recipient's gender.
Prior work relationship	Dichotomous: 1 = source and recipient have worked together, 0 = otherwise.

($\alpha = 0.95$), so we averaged the three items together to yield one measure of performance improvement.

The knowledge transfer mechanism portfolio choices include intensity and composition. Our measure of *Portfolio intensity* captures the frequency of knowledge transfers about SPI in each dyad's portfolio and is assessed by counting the number of knowledge transfers from a particular source to a particular recipient over the time period of the study. Our measure of *Portfolio composition* assesses the proportion of directions versus routines in a dyad's portfolio of knowledge transfers, and is expressed as the ratio of the number of directions in the dyad's portfolio divided by the total number of knowledge transfers in the dyad's portfolio. For example, if the portfolio of knowledge transfers for a source-recipient dyad includes eight transfers, and two of these transfers use directions while six use routines the intensity of the dyad's portfolio is 8, and it is composed of 25% (2/8) directions.

The antecedents of knowledge transfer mechanism portfolio choices include the nature of the relationship, the proximity, and the work units of each source-recipient dyad. The *Nature of the relationship* between the source and recipient is a dichotomous variable that is set to "1" if the source and recipient are in a hierarchical relationship (i.e., supervisor-subordinate or subordinate-supervisor), and "0" otherwise. We measured the *Proximity* between each source and recipient dyad using a scaled, architectural rendering of the IS department's floor plan. The distance between a source and recipient was measured by counting the number of cubicles between them, following the corridors between room sections in the floor of the building where the IS department is housed. Except for senior managers (whose cubicles are twice the size of those for other personnel), all cubicles in the department are square and of the same length; thus, counting cubicles measures the actual physical distance between the source and recipient in each dyad. Measuring proximity in terms of the actual distance between sources and recipients is consistent with literature on communication (e.g., Allen 1977). The *Work unit* construct is measured using a dichotomous variable that is set to "1" if the source and recipient are in different work units and is "0" if they are in the same work unit.

Analysis and Results

Our coding identified 1,395 knowledge transfers between 355 unique source-recipient dyads. On average, there were 3.93 knowledge transfers per dyad (computed as the average of the total number of knowledge transfers between each of the 355 dyads), with a minimum of 1 transfer and a maximum of 11 transfers per dyad. In terms of the composition, the average portfolio consists of 73% directions and 27% routines. Some source-recipient dyads (178) use portfolios with no routines and 100% directions; some dyads (33) use portfolios with 100% routines and no directions; the remaining dyads (144) use a mix of directions and routines in their portfolios. Table 3 shows the descriptive statistics and correlations for the variables in our analysis.

We analyzed the data using a simultaneous equations framework of the form: $y_{ij} = \mathbf{Y}_{ij}\gamma + \mathbf{X}_{1ij}\beta + \mu_i + \nu_{ij}$, where y_{ij} is the dependent variable (the performance improvement of recipient j in source-recipient dyad ij), \mathbf{Y}_{ij} is a vector of observations for the endogenous variables (the intensity and the composition of knowledge transfer portfolios selected by source i and recipient j), which are, in turn, predicted by \mathbf{X}_{1ij} —a vector of observations for the exogenous variables (the nature of the relationship, work unit, and proximity of source i and recipient j and the control variables), and γ and β are vectors of coefficients to be estimated in the different equations. The equation for performance predicts the performance improvement of each recipient in a source-recipient dyad based on the dyad's knowledge transfer portfolio choices for SPI: $Performance\ improvement_{ij} = \gamma_0 + \gamma_1(Portfolio\ intensity_{ij}) + \gamma_2(Portfolio\ composition_{ij}) + \gamma_3(Portfolio\ intensity_{ij} * Portfolio\ composition_{ij}) + \mu_i + \nu_{ij}$. The equations for knowledge transfer portfolio choices predict the intensity and composition of the portfolios selected by source-recipient dyads, based on the nature of the relationship between the source and recipient, the work units of the source and recipient, and the proximity of the source and recipient, and controlling for *source* and *recipient* demographics: $Portfolio\ intensity_{ij}$ or $Portfolio\ composition_{ij} = \beta_0 + \beta_1(IS\ experience_i) + \beta_2(Org\ tenure_i) + \beta_3(Education_i) + \beta_4(Gender_i) + \beta_5(IS\ experience_j) + \beta_6(Org\ tenure_j) + \beta_7(Education_j) + \beta_8(Gender_j) + \beta_9(Prior\ work_{ij}) +$

Table 3 Descriptive Statistics and Correlations

Variable name	Mean	Std. dev.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
1. Portfolio intensity, composition	3.93 0.73	3.07 0.34	-0.23** -0.23**											
2. Proximity	29.65	31.85	0.47** -0.41**	1.00										
3. Nature of relationship ^a	0.39	0.49	0.16** -0.14*	0.33**	1.00									
4. Work unit ^b	0.52	0.50	0.05 0.42**	-0.51**	-0.82**	1.00								
5. IS experience (source)	10.28	10.20	0.11* -0.08	0.45**	0.69**	-0.57**	1.00							
6. Org tenure (source)	7.82	6.54	-0.10 0.01	0.50**	0.47**	-0.48**	0.76**	1.00						
7. Education ^c (source)	3.09	0.49	-0.03 0.40**	-0.11*	-0.39**	0.44**	-0.11*	0.15*	1.00					
8. Gender ^d (source)	0.34	0.47	0.42** -0.21**	0.13*	0.11*	-0.01	0.14*	-0.02	-0.18**	1.00				
9. IS experience (recipient)	12.71	8.53	0.18** 0.06	0.11*	0.16**	-0.11*	0.08	0.03	-0.08	0.10 [†]	1.00			
10. Org tenure (recipient)	8.16	5.60	0.12* 0.06	0.08	0.14*	-0.08	0.10 [†]	0.06	-0.08	0.07	0.62**	1.00		
11. Education ^c (recipient)	2.85	0.74	-0.01 -0.06	-0.02	-0.01	-0.04	0.01	0.02	-0.03	-0.07	-0.22**	-0.29**	1.00	
12. Gender ^d (recipient)	0.38	0.49	0.01 -0.01	-0.06	-0.02	-0.02	-0.03	0.01	0.02	0.01	-0.05	-0.14*	0.08	1.00
13. Prior work	0.03	0.17	-0.19** -0.30**	0.16*	-0.14*	-0.18**	-0.14*	-0.01	-0.04	-0.13*	-0.07	-0.08	0.01	-0.01
14. Performance	4.06	1.77	-0.14** -0.08	-0.08	-0.12*	0.10 [†]	-0.07	-0.05	0.06	-0.14*	-0.25**	-0.22**	0.01	-0.24** -0.25**

Notes. $n = 355$. [†] $p < 0.10$; * $p < 0.05$; ** $p < 0.01$. Coding: ^a1 = hierarchical, 0 = nonhierarchical; ^b1 = different unit, 0 = same unit; ^c1 = high school, 2 = associate's, 3 = bachelor's, 4 = master's, 5 = Ph.D.; ^d1 = female, 0 = male. Pearson correlations are reported between pairs of continuous variables, Spearman correlations between pairs of continuous and dichotomous variables, and Phi correlations between pairs of dichotomous variables.

$$\beta_{10}(\text{Nature of relationship}_{ij}) + \beta_{11}(\text{Work unit}_{ij}) + \beta_{12}(\text{Proximity}_{ij}) + \delta_i + \epsilon_{ij}.$$

The natural logarithm of portfolio intensity is used in the portfolio intensity equation, as suggested by the Box-Cox procedure (1964) to achieve a more symmetrical distribution of the error terms. Also, every variable was standardized to its Z-score before entering it into the analysis to ease interpretation of estimated coefficients (as each variable is measured using a different unit scale), and to mitigate potential collinearity between the main effects of portfolio intensity and composition and their interactions (Aiken and West 1991). Finally, we added random effects to the equations to address the nested data, i.e., the possible correlation of residuals across dyads with the same source but different recipients or same recipi-

ent but different sources. As noted, the knowledge transfer mechanism portfolio choices are endogenous in this formulation. Thus we used a generalized two-stage least squares (G2SLS) random effects instrumental variables regression procedure to estimate the system of equations for the data panel (Greene 2003).

Diagnostics suggested no problems with multicollinearity in any of the equations, as the condition indices and variance inflation factors are all less than 10, well within acceptance levels. In addition, we found no outliers in the Performance improvement equation, five outliers in the Portfolio composition equation, and three outliers in the Portfolio intensity equation. However, the results do not change materially when the analyses are conducted without the outlying observations, and we thus report the

Table 4 Results for Portfolio Intensity and Portfolio Composition

Variable name	Portfolio intensity		Portfolio composition	
	Coefficient (Std. error)	Z-value	Coefficient (Std. error)	Z-value
Constant [β_0]	0.054 (0.020)	2.74**	-0.200 (0.017)	-11.99***
IS experience (source) [β_1]	0.184 (0.050)	3.67***	-0.089 (0.035)	-2.50*
Org tenure (source) [β_2]	-0.547 (0.049)	-11.16***	0.331 (0.029)	11.36***
Education (source) [β_3]	-0.028 (0.025)	-1.14	0.303 (0.024)	12.80***
Gender (source) [β_4]	0.147 (0.019)	7.77***	-0.062 (0.018)	-3.35**
IS experience (recipient) [β_5]	0.020 (0.018)	1.08	0.057 (0.015)	3.90***
Org tenure (recipient) [β_6]	-0.002 (0.021)	-0.07	-0.030 (0.017)	-1.74†
Education (recipient) [β_7]	0.018 (0.017)	1.04	-0.004 (0.013)	-0.27
Gender (recipient) [β_8]	0.038 (0.016)	2.44*	-0.006 (0.014)	-0.41
Prior work [β_9]	-0.005 (0.022)	-0.24	0.045 (0.022)	2.03*
Nature of relationship [β_{10}]	0.836 (0.049)	17.11***	1.067 (0.051)	21.01***
Work unit [β_{11}]	0.696 (0.047)	14.70***	1.165 (0.051)	22.71***
Proximity [β_{12}]	0.787 (0.023)	33.73***	-0.390 (0.023)	-16.62***
Deviance difference (χ^2)	5,045.97***		7,400.34***	
Pseudo- R^2	0.670		0.615	

Notes. $n = 355$. † $p < 0.10$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$. Dependent variable is the natural logarithm of portfolio intensity in the portfolio intensity equation. Estimates obtained via generalized least squares regression (with a random effect for each source and recipient in a dyad and a correction for heteroskedastic panels).

results with all observations included. Finally, heteroskedastic error structures were found in the Portfolio composition and Portfolio intensity equations, and we corrected for heteroskedasticity in the estimation of those equations. Table 4 presents the statistical results from the estimation of the equations for portfolio intensity and portfolio composition.

The results suggest strong support for our hypotheses predicting the intensity and composition of knowledge transfer portfolio choices. As shown in Table 4, H1A is supported as a more intense portfolio is used when the source and recipient are in differ-

ent work units ($\beta_{11} = 0.70$, $z = 14.70$, $p < 0.01$). Similarly, H1B is supported as more directions are used in the portfolio when the source and recipient are in different work units ($\beta_{11} = 1.17$, $z = 22.71$, $p < 0.01$). Table 4 also shows that H2A is supported as a source and recipient in a hierarchical relationship choose a more intense knowledge transfer portfolio than a dyad not in such a relationship ($\beta_{10} = 0.84$, $z = 17.11$, $p < 0.01$). H2B is supported as a source and recipient in a hierarchical relationship use more directions in their knowledge transfer portfolio ($\beta_{10} = 1.07$, $z = 21.01$, $p < 0.01$). Finally, H3A is supported as a more proximate source and recipient transfer knowledge more intensely than a source and recipient who are less proximate ($\beta_{12} = 0.79$, $z = 33.73$, $p < 0.01$). A more proximate source and recipient also use more routines in their knowledge transfer portfolio as predicted in H3B ($\beta_{12} = -0.39$, $z = -16.62$, $p < 0.01$). Together, the nature of the relationship, proximity, and work units of the source and recipient explain substantial variation in the design of knowledge transfer portfolios (explaining 30% of the variation in the intensity of the portfolio selected, $\chi^2 = 342.30$, $p < 0.01$; explaining 35% of the variation in the composition of the portfolio selected, $\chi^2 = 234.02$, $p < 0.01$).

Our final set of hypotheses concerns the performance effects of knowledge transfer portfolio choices. Table 5 shows the results. To evaluate H4, we differentiated the Performance improvement equation with respect to portfolio intensity, and examined the change in performance improvement with a unit change in intensity, holding portfolio composition constant at

Table 5 Results for Performance Improvement

Variable name	Coefficient (Std. error)	Z-value
Constant [γ_0]	3.963 (0.356)	11.14***
Portfolio intensity [γ_1]	1.422 (0.576)	2.47*
Portfolio composition [γ_2]	0.770 (0.479)	1.61†
Portfolio intensity × Portfolio composition [γ_3]	-2.560 (0.740)	-3.46***
Deviance difference (χ^2)	17.65***	
Pseudo- R^2	0.380	

Notes. $n = 355$. † $p < 0.10$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$. Estimates obtained via G2SLS regression (with a random effect for each source and recipient in a dyad).

its mean value (Greene 2003). We find that H4 is supported as a recipient of a more intense knowledge transfer portfolio has a greater performance gain from SPI than a recipient of a less intense portfolio, when evaluated at an average use of directions in the portfolio ($\partial \text{Performance} / \partial \text{Portfolio intensity} = \gamma_1 + \gamma_3(\text{Portfolio composition}_{ij}) = \gamma_1$ evaluated at the mean value for *portfolio composition* = 1.42, $z = 2.47$, $p < 0.05$).

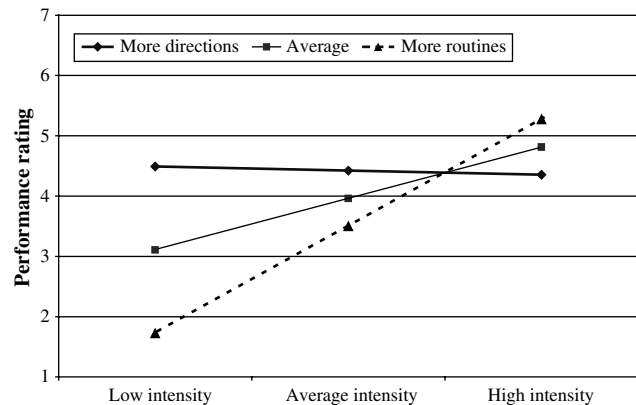
Using a similar approach, we evaluate the main effect of portfolio composition (H5). We find support, albeit somewhat weaker, for this hypothesis as a recipient of a knowledge transfer portfolio with more directions achieves a greater performance gain from SPI than a recipient of a portfolio with more routines, when evaluated at average levels of intensity in the portfolio ($\partial \text{Performance} / \partial \text{Portfolio composition} = \gamma_2 + \gamma_3(\text{Portfolio intensity}_{ij}) = \gamma_2$ when evaluated at the mean value of zero for *portfolio intensity* = 0.77, $z = 1.61$, $p < 0.10$). Finally, the coefficient of the interaction of portfolio intensity and composition is negative and significant ($\gamma_3 = -2.56$, $z = -3.46$, $p < 0.001$), suggesting support for H6. We discuss our findings and their implications next.

Discussion

Several important insights can be gleaned from our results. First and foremost, our findings indicate that *how* organizational members transfer software process knowledge affects the level of individual performance gains from the SPI initiative. Both intensity and composition are significant dimensions of knowledge transfer portfolios, and some combinations of intensity and composition are more effective than others. Figure 3 graphs the interaction between portfolio intensity and composition. As shown in Figure 3, with greater portfolio intensity, a portfolio with a higher proportion of routines results in higher performance gains, while at lower levels of portfolio intensity, a portfolio with a higher proportion of directions is more effective.¹

¹ Note that “low” and “high” intensity correspond to one standard deviation below and above the mean intensity of the portfolio, respectively. For directions, “more directions” corresponds to one standard deviation above the mean percentage of directions in the portfolio, while “more routines” corresponds to one standard deviation below the mean percentage of directions in the portfolio.

Figure 3 Portfolio Intensity × Composition Interaction



For our sample, a sensitivity analysis suggests that at portfolio intensity equal to 4.85 transfers per dyad (the point on the graph where all three lines intersect), a portfolio composed of 63% directions and 37% routines is effective (the average portfolio has 73% directions and 27% routines). However, when portfolio intensity is below or above 4.85 transfers per dyad, then different compositions of directions and routines are more effective. When portfolio intensity is greater than 4.85 transfers per dyad, portfolios composed of more routines (>37% routines) are more effective, while at portfolio intensity less than 4.85 transfers per dyad, portfolios composed of more directions (>63% directions) are more effective. Note also, that in our sample, the highest performance gains (performance rating of seven) are achieved for dyads whose portfolios are composed of 100% routines, with an intensity level of seven knowledge transfers. The highest performance rating that can be achieved for dyads whose portfolios are composed of 100% directions is 5.86 (this occurs when their portfolio has the lowest intensity level of one transfer).

As Grant (1996) has suggested, directions are more efficient mechanisms for knowledge transfer because routines require significant amounts of repetition and interaction before they can be used effectively. Thus, it would follow that, at low portfolio intensity, directions have a greater impact than routines, and this is consistent with our sensitivity analyses. However, to achieve higher levels of performance requires portfolios with a larger proportion of routines as well as greater intensity. Our finding that

the highest performance gains ensue from portfolios with 100% routines and a high level of intensity suggests that to achieve significant performance improvements requires a more in-depth understanding, application, and “embedding” of software process improvements in work practices (Ravichandran and Rai 2003), and that this embedding is more effectively realized when high intensity portfolios composed of routines are used.

Our analysis of performance suggests the importance of knowledge transfer mechanisms for software process improvements. It also reveals the optimal combinations of intensity and composition in dyads’ knowledge transfer portfolios to maximize the performance gains from SPI. A natural follow-on question is to ask how to achieve these optimal levels of portfolio design. Our analysis of the antecedents of portfolio design choices provides insight into this question. In particular, we find that organizational design choices strongly affect the nature of the knowledge transfer portfolios used by dyads. Our results indicate that a more intense portfolio of knowledge transfer mechanisms is used in SPI when the source and recipient are proximate, when they are in a hierarchical relationship or when they work in different units. In addition, we find that more directions are used in SPI when the source and recipient are farther apart, when they are in a hierarchical relationship, or when they work in different units.

Our findings highlight the role that organizational design decisions play in the transfer of knowledge that is complex and contextual such as in SPI. Linking our results from the performance and portfolio design equations reveals how organizational design decisions can be structured to yield higher performance gains from SPI. Table 6 shows how organizational design decisions in this firm relate to portfolio choices and ultimate performance impacts. The optimal cells in the table indicating the highest performance gains are highlighted in bold borders. Recall from our sensitivity analysis that portfolios lower in intensity (<4.85 transfers) and high in composition (>63% directions), or high in intensity (>4.85 transfers) and low in composition (>37% routines), are most effective. That is, high performance can be achieved at lower intensity with a portfolio composed

Table 6 Integrated Results

Portfolio composition	Portfolio intensity	
	Low (<4.85 transfers)	High (>4.85 transfers)
More directions (<37% routines)	Nonproximate dyad	Hierarchical relationship Different work units
More routines (>37% routines)	Nonhierarchical relationship Same work unit	Proximate dyad

of more directions or at higher intensity with a portfolio composed of more routines.

Also shown in Table 6 are the organizational design choices (i.e., proximity, hierarchy, work unit structure) that correspond to the intensity and composition levels of the portfolio. Table 6 suggests that, for our sample, the only design choice that reflects an optimal knowledge transfer portfolio structure (in terms of intensity and composition) is proximity: distant source-recipient dyads use directions infrequently, while proximate source-recipient dyads use routines frequently. This suggests that proximity facilitates the choice of a portfolio that achieves composition—intensity *fit*, which, in turn, has a positive impact on performance.

In contrast, those dyads in a hierarchical relationship, or those who work in different units, use too much intensity given the level of directions in their portfolio composition, as seen in Table 6. That is, these sources are transferring knowledge via directions in an intense manner, but this portfolio design is inefficient in terms of performance impact. For example, sources may be issuing frequent memos (directions) more often than is required to transfer the knowledge. The impact of this portfolio on the performance improvement of the recipient is not as successful as it could be. Table 6 suggests that to increase performance, dyads in a hierarchical relationship or in different work units can reduce the frequency of their knowledge transfers, given the high proportion of directions in their portfolio.

Similarly, Table 6 indicates that dyads who are peers, or who work in the same unit, use routines but too infrequently, resulting in suboptimal performance improvement. For example, a developer on

one project team may make a one-time demonstration of an improved design technique to another developer on a different project team. However, because the recipient developer does not make frequent observations, he does not gain an in-depth understanding of the knowledge, and thus does not see many gains in his performance. Table 6 suggests that to increase performance, dyads that are peers or in the same work unit should increase the intensity of their knowledge transfers, given the high level of routines in their portfolio.

SPI has been identified as an activity that is complicated and difficult, and it can require years of effort for an organization to advance in software capability maturity (e.g., Harter et al. 2000). Further, SPI is an initiative that often fails to achieve its objectives; Hardgrave and Armstrong (2005) cite failure rates of up to 70%. Our findings suggest a practical strategy for improving the performance gains from SPI. First, the performance impact from SPI should be measured after some period of time. Managers should then identify where the highest performance gains are occurring and relate these gains to the knowledge transfer portfolios used to achieve them. Finally, managers should make any necessary modifications to organizational design parameters to facilitate the selection of those portfolios.

Conclusions

Our objective in this study is to understand how organizations can be more successful in improving their software development processes. We approached this question from the perspective of knowledge transfer, considering how software process knowledge is transferred between source and recipient dyads within an IS department, and relating how knowledge is transferred to the recipients' performance gains from process improvement. In concluding, we note the contributions and limitations of this study and suggest directions for future research.

Our study theorizes and provides empirical evidence about the design choices for a *portfolio* of knowledge transfer mechanisms. This focus on portfolios offers a unique and important contribution to the IS literature because it highlights how dyads utilize a *set* of mechanisms to transfer knowledge for SPI.

We identified intensity and composition as two significant dimensions of knowledge transfer portfolios, using the typology offered by Grant (1996) to analyze portfolio composition. While this typology is theoretically based and is well suited to the SPI context, it does constrain the categorization of mechanisms into directions or routines. Future research could explore other dimensions of knowledge transfer portfolios, such as the ratio of formal to informal mechanisms or of codified to personalized mechanisms, or the timing of the mechanisms deployed in the portfolio. Other contexts should be studied as well.

In addition, our study provides insight into the effectiveness of different knowledge transfer portfolio designs. Because the ultimate goal of SPI is to improve individual and organizational performance, linking performance to portfolio design choices demonstrates the importance of knowledge transfer in this context. Further, our results imply that the fit between the composition and intensity of a dyad's knowledge transfer portfolio impacts performance improvement. Thus our analysis not only demonstrates the significance of studying knowledge transfer portfolios, but also suggests the importance of conducting further research to deepen our understanding of the effectiveness of different knowledge transfer portfolio choices.

Finally, our findings highlight the importance of organizational design decisions in influencing choices of knowledge transfer portfolio intensity and composition, and thereby performance. While proximity, nature of relationship, and work units were found to be influential in our study, there are undoubtedly other organizational design parameters that could influence portfolio choices such as the span of control. Future research is needed to determine the most effective combination of mechanisms in a portfolio, given differences in organizational design and workplace characteristics. Individual differences could also motivate the choice of different portfolios, but such factors may be more challenging for managers to influence.

Further research on other aspects of knowledge transfer portfolios would be instructive. For example, we were not able in a field setting to isolate and measure different dimensions of the knowledge transferred (such as amount and type). Experimental studies that control for the nature of the knowledge being transferred may be helpful to confirm and extend the

findings from the field. In addition, some research has suggested that politics can play a role in knowledge transfer (e.g., Carlile 2004). However, our study was not designed to capture data about power relations, and we cannot discern how political considerations may have influenced knowledge transfer in this context. This would be an interesting direction for future research.

The implications of our findings for SPI transcend discrete software projects and instead speak to the need for a deep holistic understanding of software practices and processes that span individuals and their immediate responsibilities. Our findings highlight the importance of carefully designing IS departments so that knowledge is transferred effectively across roles and units, facilitating the improvement of software processes for the department as a whole. Additional research targeted at this metalevel can yield further insights into how IS departments can continually improve their ability to consistently build and deploy software that adds value to their firms.

Acknowledgments

Prior versions of this research were presented in research seminars at the University of Pittsburgh, University of Minnesota, and Carnegie Mellon University. The authors gratefully acknowledge the research assistance of Ranjit Tinaikar and Mark Haney of the University of Pittsburgh for their help with data coding. The authors thank Ronald Slaughter for his detailed architectural rendering of the IS Department's floor plan. They are also grateful to the senior editor, associate editor, and anonymous reviewers who provided helpful comments and suggestions on earlier versions of this manuscript.

References

Adler, P. 1990. Shared learning. *Management Sci.* 36(8) 938–957.

Aiken, L., S. West. 1991. *Multiple Regression: Testing and Interpreting Interactions*. Sage Publications, Thousand Oaks, CA.

Alavi, M. 2000. Managing organizational knowledge. R. W. Zmud, ed. *Framing the Domains of IT Management*. Pinnaflex Educational Resources, Cincinnati, OH, 15–28.

Allen, T. 1977. *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information within the R&D Organization*. MIT Press, Cambridge, MA.

Argote, L. 1993. Group and organizational learning curves: Individual, system and environmental components. *British J. Soc. Psych.* 32 31–51.

Argote, L. 1999. *Organizational Learning: Creating, Retaining and Transferring Knowledge*. Kluwer Academic Publishers, Boston, MA.

Argote, L., P. Ingram. 2000. Knowledge transfer: A basis for competitive advantage in firms. *Organ. Behav. Human Decision Processes* 82(1) 150–169.

Birkinshaw, J., R. Nobel, J. Ridderstråle. 2002. Knowledge as a contingency variable: Do the characteristics of knowledge predict organization structure? *Organ. Sci.* 13(3) 274–289.

Box, G., D. Cox. 1964. An analysis of transformations. *J. Roy. Statist. Soc. B* 26 211–243.

Carlile, P. 2004. Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries. *Organ. Sci.* 15(5) 555–568.

Contu, A., H. Willmott. 2003. Re-embedding situatedness: The importance of power relations in learning theory. *Organ. Sci.* 14(3) 283–296.

Crampton, S., J. Wagner. 1994. Percept-percept inflation in microorganizational research: An investigation of prevalence and effect. *J. Appl. Psych.* 79(1) 67–76.

Crowston, K., E. Kammerer. 1998. Coordination and collective mind in software requirements development. *IBM Systems J.* 37(2) 227–245.

Darr, E., L. Argote, D. Epple. 1995. The acquisition, transfer, and depreciation of knowledge in service organizations. *Management Sci.* 41(11) 1750–1762.

Demsetz, H. 1991. The theory of the firm revisited. O. Williamson, S. Winter, eds. *The Nature of the Firm*. Oxford University Press, New York, 159–178.

Epple, D., L. Argote, R. Devadas. 1991. Organizational learning curves: A method for investigating intra-plant transfer of knowledge acquired through learning by doing. *Organ. Sci.* 2(1) 58–70.

Galbraith, C. 1990. Transferring core manufacturing technologies in high technology firms. *California Management Rev.* 32(4) 56–70.

Gaudin, S. 2003. Study: Many major IT projects still fail. *Data-mation* (June 16). http://www.itmanagement.earthweb.com/it_res/article.php/2222391.

Grady, R. 1997. *Successful Software Process Improvement*. Prentice-Hall, Upper Saddle River, NJ.

Grant, R. 1996. Prospering in dynamically-competitive environments: Organizational capability as knowledge integration. *Organ. Sci.* 7(4) 375–387.

Greene, W. 2003. *Econometric Analysis*, 3rd ed. Prentice-Hall, Upper Saddle River, NJ.

Grover, V., T. Davenport. 2001. General perspectives on knowledge management: Fostering a research agenda. *J. MIS* 18(1) 5–21.

Hansen, M. 1999. The search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Admin. Sci. Quart.* 44(1) 82–111.

Hardgrave, W., D. Armstrong. 2005. Software process improvement: It's a journey, not a destination. *Comm. ACM* 48(11) 93–96.

Harter, D., M. Krishnan, S. Slaughter. 2000. Effects of process maturity on quality, cost and cycle time in software product development. *Management Sci.* 46(4) 451–466.

Herbsleb, J., D. Zubrow, D. Goldenson, W. Hayes, M. Paulk. 1997. Software quality and the capability maturity model. *Comm. ACM* 40(6) 30–40.

Hinds, P., S. Kiesler. 1995. Communication across boundaries: Work, structure, and use of communication technologies in a large organization. *Organ. Sci.* 6(4) 373–393.

- Ko, D., L. Kirsch, W. King. 2005. Antecedents of knowledge transfer from consultants to clients in enterprise system implementation. *MIS Quart.* **29**(1) 59–85.
- Kraut, R., L. Streeter. 1995. Coordination in software development. *Comm. ACM* **38**(3) 69–81.
- Malhotra, A., A. Majchrzak, R. Carman, V. Lott. 2001. Radical innovation without collocation: A case study at Boeing-Rocketdyne. *MIS Quart.* **25**(2) 229–249.
- Mathiassen, L., P. Pourkomeylian. 2003. Managing knowledge in a software organization. *J. Knowledge Management* **7**(2) 63–80.
- McGrath, J., L. Argote. 2001. Group processes in organizational contexts. M. Hogg, R. Tindale, eds. *Blackwell Handbook of Social Psychology: Group Processes*. Blackwell Publishers, Malden, MA, 603–627.
- Mitki, Y., A. Shani, Z. Meiri. 1997. Organizational learning mechanisms and continuous improvement: A longitudinal study. *J. Organ. Change Management* **10**(5) 426–446.
- Nelson, K., J. Coopridge. 1996. The contribution of shared knowledge to IS group performance. *MIS Quart.* **20**(4) 409–432.
- Nonaka, I. 1994. A dynamic theory of organizational knowledge creation. *Organ. Sci.* **5**(1) 14–37.
- Nonaka, I., N. Konno. 1998. The concept of “ba”: Building a foundation for knowledge creation. *California Management Rev.* **40**(3) 40–54.
- Olson, G., J. Olson. 2000. Distance matters. *Human-Comput. Interaction* **15**(2/3) 139–178.
- Paulus, P., H.-C. Yang. 2000. Idea generation in groups: A basis for creativity in organizations. *Organ. Behav. Human Decision Processes* **82**(1) 76–87.
- Ravichandran, T., A. Rai. 2000. Quality management in systems development: An organizational system perspective. *MIS Quart.* **24**(3) 381–415.
- Ravichandran, T., A. Rai. 2003. Structural analysis of the impact of knowledge creation and knowledge embedding on software process capability. *IEEE Trans. Engrg. Management* **50**(3) 270–284.
- Rulke, D., S. Zaheer, M. Anderson. 2000. Sources of managers’ knowledge of organizational capabilities. *Organ. Behav. Human Decision Processes* **82**(1) 134–149.
- Schulz, M. 2001. The uncertain relevance of newness: Organizational learning and knowledge flows. *Acad. Management J.* **44**(4) 661–681.
- Spencer, B. A. 1994. Models of organization and total quality management: A comparison and critical evaluation. *Acad. Management Rev.* **19**(3) 446–471.
- Swanson, E., C. Beath. 1990. Departmentalization in software development and maintenance. *Comm. ACM* **33**(6) 658–667.
- Tashakkori, A., C. Teddlie. 1998. *Mixed Methodology: Combining Qualitative and Quantitative Approaches*. Sage Publications, Thousand Oaks, CA.
- Teasley, S., L. Covi, M. Krishnan, J. Olson. 2002. Rapid software development through team collocation. *IEEE Trans. Software Engrg.* **28**(7) 671–691.
- Thomas-Hunt, M., T. Ogden, M. Neale. 2003. Who’s really sharing? Effects of social and expert status on knowledge exchange within groups. *Management Sci.* **49**(4) 464–477.