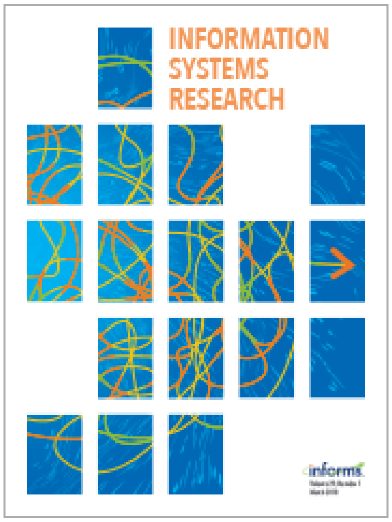

Information Systems Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Learning Effects of Domain, Technology, and Customer Knowledge in Information Systems Development: An Empirical Study

Keumseok Kang, Jungpil Hahn, Prabuddha De

To cite this article:
Keumseok Kang, Jungpil Hahn, Prabuddha De (2017) Learning Effects of Domain, Technology, and Customer Knowledge in Information Systems Development: An Empirical Study. Information Systems Research 28(4):797-811. https://doi.org/10.1287/isre.2017.0713

informs
http://pubsonline.informs.org/journal/isre/

INFORMATION SYSTEMS RESEARCH
Vol. 28, No. 4, December 2017, pp. 797–811
ISSN 1047-7047 (print), ISSN 1526-5536 (online)

Learning Effects of Domain, Technology, and Customer Knowledge in Information Systems Development: An Empirical Study

Keumseok Kang,[a] Jungpil Hahn,[b] Prabuddha De[c]

[a] Department of Information Systems and Business Analytics, College of Business, Florida International University, Miami, Florida 33199;
[b] Department of Information Systems and Analytics, School of Computing, National University of Singapore, Singapore 117417;
[c] Department of Management, Krannert School of Management, Purdue University, West Lafayette, Indiana 47907

Contact: kskang@fiu.edu, http://orcid.org/0000-0001-9034-5712 (KK); jungpil@nus.edu.sg, http://orcid.org/0000-0001-8388-4865 (JH); pde@purdue.edu, http://orcid.org/0000-0002-0881-9294 (PD)

Received: July 29, 2015
Revised: April 25, 2016; December 8, 2016; February 26, 2017
Accepted: March 17, 2017
Published Online in Articles in Advance: October 20, 2017

https://doi.org/10.1287/isre.2017.0713

Abstract. This study examines learning effects (i.e., the effects of prior experience) in information systems development (ISD). ISD is characterized by disparate tasks, teams, and levels of project complexity across projects. These features challenge our understanding of how learning effects occur in the ISD context. Drawing on the theory of transfer of learning, this study examines how ISD project teams learn and under what conditions the learning effects are stronger or weaker. We find that ISD project teams' experience in prior projects translates into performance gains for the current ISD project when the prior and current projects share the same domain, technology, or customer knowledge elements—domain, technology, and customer being the most essential knowledge types for ISD. Moreover, we find that the learning effects of domain, technology, and customer knowledge are substitutive for one another and that these learning effects become stronger or weaker depending on the extent of ISD projects' team and task complexities. The study makes significant contributions to the ISD literature on learning effects and the roles of domain, technology, customer knowledge, and project complexity, as well as to the general organizational learning literature. It also provides important managerial insights into practical concerns such as project staffing and knowledge acquisition for ISD organizations.

History: Sabyasachi Mitra, Senior Editor; Giri Kumar Tayi, Associate Editor.
Funding: This study was supported by the National University of Singapore Academic Research Fund (AcRF) Tier 1 Start Up Research Grant [R-253-000-098-133].
Supplemental Material: The online appendix is available at https://doi.org/10.1287/isre.2017.0713.

Keywords: information systems development • learning effects • domain knowledge • technology knowledge • customer knowledge • project complexity • empirical research

1. Introduction

Despite decades of advances in programming languages, development methods and tools, and formal training in computer science and information systems, it is the unfortunate reality that information systems development (ISD) is still plagued by performance concerns such as frequent cost and/or schedule overruns (Brooks 1995, Rubenstein 2007). Although success rates have gradually improved, the majority of ISD projects face challenges in successfully delivering intended systems (Ewusi-Mensah 2003, El Emam and Koru 2008).

While there may be many different factors leading to low success rates of ISD, a direct investigation of ISD performance by identifying the factors associated with its increase (or decrease) has recently been the focus of much attention in the information systems literature. Most notably, many researchers have begun to adopt a knowledge-based learning perspective to propose that ISD performance can be improved by taking advantage of learning curve (or experience curve) effects (e.g., Boh et al. 2007, Narayanan et al. 2009).

The basic idea behind learning curve effects (or simply, learning effects) is quite straightforward—repeated experiences (or trials) create a growing stock of knowledge that improves performance (Argote 2013). According to the theory of transfer of learning (Ellis 1965, Thorndike and Woodworth 1901, Perkins and Salomon 1994, Schunk 2015), the main ingredient of learning curves is the existence of the *same elements* between prior experience and the current task (i.e., across trials). In this regard, learning effects are typically examined in a setting where delivered products (or services) are identical, and the unit of the same experience (i.e., the same repeated task) is defined at the level of products (or services). Most learning curve studies in the software engineering and information systems context (e.g., Boh et al. 2007, Narayanan et al. 2009) also follow similar conventions. They measure ISD learning effects in the information systems maintenance (ISM) context where the *same system* is maintained and use *the system or subsystem* to represent the same experience (e.g., prior experience working on the same system or subsystem).

Although this choice of the unit of the same experience is appropriate for the ISM context, this may not be the case in the context of ISD where a software service company executes a variety of ISD projects requested by different customers.[1] Although ISD shares many commonalities with ISM, it also includes several unique characteristics that will inevitably affect learning effects. A review of the literature highlights the following as the unique characteristics of ISD (as compared to ISM): disparate tasks (i.e., each ISD project develops a different information system) (Huckman et al. 2009), disparate teams (i.e., each ISD project comprises a different set of team members) (Huckman and Staats 2011), and disparate complexity (i.e., each ISD project incorporates a different level of complexity) (Xia and Lee 2005). These are important and salient factors to be considered when studying learning effects in the ISD context.

Given that effective transfer of learning is more likely to occur between the same task and within the same learner (Ellis 1965, Thorndike and Woodworth 1901, Perkins and Salomon 1994), the disparate nature of ISD in the above dimensions creates challenges for transfer of learning to occur. Does this mean that learning effects would not exist in the ISD context? We believe the answer is "no." This conjecture is based on the observation that when we look closely at the systems developed in ISD, they are not totally different, but share common elements across projects. This suggests that learning effects should exist in ISD as well. The question is *how* (*i.e., in what form*) *learning effects exist* in ISD.

Huckman and colleagues (2009) offer empirical results that provide a good starting point for further investigation into this question. They find that in ISD learning effects are observed when the project member role (i.e., experience working in the same role such as project manager or nonmanager in prior projects) is used as the unit of the same experience. Indeed, role experience is an important and novel way to conceptualize repeated experiences in ISD. However, given the multidimensional nature of ISD projects, further research is needed to characterize the same experience in other dimensions and to uncover associated learning effects to extend our understanding of how learning occurs in ISD.

In this study, we propose that characterizing the same experience using the knowledge required is a useful and intuitive way to understand learning effects in ISD. ISD is knowledge-intensive work that requires the application of various knowledge areas (Langer et al. 2014, Lee et al. 1995, Jiang et al. 1998). In particular, the ISD literature finds *domain*, *technology*, and *customer* as the most essential knowledge areas (Gopal and Gosain 2010, Khatri et al. 2006, Huckman and Staats 2011, Ethiraj et al. 2005, Garud 1997). We conceptualize ISD as the combined application of domain, technology, and customer knowledge and argue that different projects may share common domains, technologies, and/or customers with one another. Hence, our first research question asks the following: Do ISD project teams benefit from prior experiences in domain, technology, and customers in increasing project performance?

Interesting follow-up questions would be whether and how experiences in the three types of knowledge relate to one another: Do they interfere with or substitute for one another so that their combined effects on performance are diminished (i.e., a negative interaction effect)? Do they complement one another to produce additional synergistic effects on performance as the level of experience in each knowledge type increases (i.e., a positive interaction effect)? Are they independent, such that their effects are simply additive in affecting ISD performance? These follow-up questions are important theoretically because the extant theory suggests that knowledge in one type may interfere with (Underwood 1957, Levinthal and March 1993) or substitute for (Argote and Ingram 2000, Walsh and Ungson 1991) the accumulation of knowledge in another type, or one type may, conversely, facilitate the acquisition of another type (Cohen and Levinthal 1990). Therefore, our second research question asks the following: Whether the learning effects of domain, technology, and customer knowledge are substitutive, complementary, or additive?

Finally, it is also known that learning is affected by the context of learning (Argote 2013). In this study, we draw on the related ISD literature (Xia and Lee 2005, Langer et al. 2014) and consider ISD project complexity as one of the most influential contextual factors for project performance (Xia and Lee 2005). However, it is not clear whether and how project complexity *substitutes, complements* or *offsets* the effects of domain, technology, and customer experience. Hence, our third research question asks the following: How does ISD project complexity moderate the learning effects of these three types of experience?

We provide answers to the above three research questions drawing on the theory of transfer of learning (Ellis 1965, Royer 1979, Day and Goldstone 2012). Using an extensive archival data set of ISD projects from a prominent international information technology (IT) services company, we observe project team-level learning effects for domain, technology, and customer knowledge. We also find that the learning effects of domain, technology, and customer knowledge are substitutive for one another and are negatively moderated by ISD project complexity.

2. Theory and Hypotheses

2.1. Organizational Learning Curves and Theory of Transfer of Learning

Repeated experiences increase organizational performance. This widely documented phenomenon is generally referred to as organizational learning curve effects or simply learning effects (Argote 2013). The theoretical underpinning for learning effects traces back to the theory of transfer of learning (Ellis 1965, Schunk 2015), which posits that overall performance improves with the number of trials of task execution because the performance of the current execution (i.e., the target task) embodies the learning transferred from a previous trial (i.e., the source task). The key is whether or not the knowledge acquired from the source task can be effectively transferred to the target task.

There are two theoretical perspectives explaining the mechanism of transfer of learning. The first is the *environmental theory of transfer of learning*, which focuses on the environmental factors of learning, that is, on the characteristics of repeated tasks—e.g., identical versus nonidentical (Thorndike and Woodworth 1901), near versus far (Perkins and Salomon 1994), lateral versus vertical (Gagné 1965), specific versus nonspecific (Ellis 1965), and literal versus figural (Royer 1979)—and argues that transfer occurs primarily because of properties of the repeated tasks. The most representative theory in this stream is identical elements theory (Thorndike and Woodworth 1901) that suggests that transfer between tasks takes place only if the tasks share identical elements. Perkins and Salomon (1994) extend this argument by distinguishing between "near tasks" where there is significant overlap between the source and target tasks and "far tasks" where the overlap is less. The extent of transfer, and hence the strength of the learning effect is positively related to the extent of overlap between tasks. Consequently, the task performance may improve not only as the cumulative experience on a particular task increases but also over experiences of disparate tasks if there is sufficient overlap across tasks.

The second perspective is the *cognitive theory of transfer of learning* (Royer 1979, Day and Goldstone 2012), which focuses on the learners' cognitive processes rather than on tasks. This perspective argues that the likelihood of learning transfer is determined by the ability of learners in retrieving relevant prior experience stored in memory. This means that transfer of learning is more likely to occur when the learner is able to capture connections between past experience and the current task. Conversely, even if similarity exists between source and target tasks, if these are cognitively difficult to identify, transfer of learning is less likely to occur and eventual learning effects will diminish. In summary, both the relationship across repeated tasks and the learner's cognitive processes affect the extent of transfer of learning. Learning effects are likely to occur when past experience and the current task share more identical elements and when this similarity between past experience and the current task is cognitively easier to identify.

2.2. Learning Effects in ISD Project Teams

The ISD context can be conceptualized as a make-to-order (or assemble-to-order) production process where the final products or services that are repeatedly produced are not completely identical, but are constructed by assembling a set of reusable elements. Each information system delivered is unique, but it often shares common business rules and technology components with other systems previously developed. In such settings, the unit of the same experience should be at the level of the reusable element rather than the final product so as to appropriately reflect learning effects (Kantor and Zangwill 1991) because here the basic unit of task repetition and transfer of learning is the reusable element (Ellis 1965). The key question is then, what is a useful conceptualization and classification of repeating elements? We propose that the types of knowledge used in ISD can serve as a useful way to capture learning effects. We conceptualize ISD as the application of various elements of domain, technology, and customer knowledge, and propose that the units of the same experience should be these knowledge elements.

Domain knowledge refers to the knowledge about the application domain of the information system being built and the context in which the system will be used (Gopal and Gosain 2010, Khatri et al. 2006). For instance, to develop a production planning system for a manufacturing company, the ISD project team must understand fundamental production-planning concepts such as bill-of-materials, capacity, safety stock, manufacturing lead time, etc.; and how these concepts fit into different types of production plans such as aggregate plans, master production schedules, material requirements plans, etc. *Technology knowledge* refers to the knowledge about technical building blocks and how to use them in the implementation of information systems (Gopal and Gosain 2010, Khatri et al. 2006). This includes knowledge relating to the various hardware and software technologies, how they work, and how to integrate them. Examples of technology knowledge include operating systems (e.g., Windows, Linux), programming languages (e.g., Java, COBOL), database technologies (e.g., ISAM, relational databases, object-oriented databases), network technologies (e.g., local area network, firewall, messaging protocols), etc. *Customer knowledge* refers to the knowledge about the organization and its users who set the requirements of the target information system and use the system (Ethiraj et al. 2005, Garud 1997, Huckman and Staats 2011). Each customer is different. Each may have a unique culture, organizational structure, operating procedures, decision-making processes, technical environments, IT capabilities, stakeholders, etc. It is important for an ISD project team to correctly understand the unique customer characteristics since the success of an ISD project is often influenced by them (Ethiraj et al. 2005, Garud 1997, Huckman and Staats 2011). Domain, technology, and customer knowledge are essential to ISD because every ISD project requires these three types of knowledge although the specific elements within the types may not be the same across projects.

Consider the following stylized example: Project P_1 is for an e-commerce application (domain) developed using the Java programming language (technology) for *HomeApplianceCo* (customer); project P_2 is also for an e-commerce application (domain) but developed using the ASP.NET programming language (technology) for *PharmaCo* (customer); and finally project P_3 is for a knowledge management application (domain) developed using the C programming language (technology) for *ConsultingCo* (customer). All three projects are, strictly speaking, different, but projects P_1 and P_2 share knowledge from the same domain (i.e., e-commerce). Although the final systems are not identical, prior experience in project P_1 may be partially beneficial to project P_2 since the projects share the common domain knowledge element. Conversely, project P_3 may not benefit from prior experience in project P_1 or P_2 since it does not share any common domain, technology, or customer knowledge elements with these projects.

According to the environmental perspective (Perkins and Salomon 1994, Thorndike and Woodworth 1901), learning effects accrue as a result of repeated execution of identical elements across tasks. Therefore, an ISD project team's prior experience, which consists of *domain experience* (i.e., experience with the same domain knowledge elements), *technology experience* (i.e., experience with the same technology knowledge elements), and *customer experience* (i.e., experience with the same customer), should lead to learning effects (i.e., performance gains).[2] Based on the discussion above, we propose the following baseline hypotheses:

Hypothesis 1A (H1A). *The ISD project team's domain experience positively impacts the project team's performance.*

Hypothesis 1B (H1B). *The ISD project team's technology experience positively impacts the project team's performance.*

Hypothesis 1C (H1C). *The ISD project team's customer experience positively impacts the project team's performance.*

2.3. Interactions Among Domain, Technology, and Customer Experiences

Now we consider how domain, technology, and customer experiences interact with one another and affect their overall learning effects.[3] We expect that domain and technology experiences will have a substitutive interaction effect. From a problem solving perspective, domain knowledge provides the ISD project team with information relevant to the problem space, whereas technology knowledge provides information relevant to the solution space (Adelson and Soloway 1985, Khatri et al. 2006). Problem and solution spaces may not always be totally independent but relate to one another. In ISD, domains and technologies may provide similar knowledge to the ISD project team. For example, e-commerce applications are typically built using web technologies (e.g., ASP.NET, JSP), where some domain functions of e-commerce applications (e.g., user authentication, user session management) are constrained or characterized by features of the web technologies used (e.g., form-based authentication, memoryless sessions). Hence, working with web technologies may provide valuable knowledge about the domain requirements of e-commerce, which can alternatively be gained from working on e-commerce applications. Similarly, working on e-commerce applications may provide developers with knowledge related to web technologies, which can also be alternatively gained from working with web technologies.

We expect that customer experience and domain or technology experience will also have substitutive interaction transfer effects. There are two reasons for this. First, although they are not identical as a whole, both experiences share some common elements that provide the same transfer effect to subsequent ISD projects. Customer experience provides the project team with information about the customer's overall functional and technical contexts where the target information system will be introduced, including the customer's business context, operating procedures, organizational structure, culture, technical capability, technical architecture, legacy systems, and users (Ethiraj et al. 2005, Garud 1997). Hence, although these may not directly provide the functional and technical requirements of the target system, customer experience may provide contextual information about the domain (e.g., business context, operating procedures, etc.) and technology (e.g., technical capability, architecture, legacy systems, etc.) of the target system, which the project team may also be able to collect and understand using past experience in the same domain and/or technology. In other words, if a project team has customer experience, the team already has some level of capability to collect and understand information about the functional and technical requirements and constraints of the target system because customer experience partially substitutes for the role of domain and technology experience. This means that, all else being equal, a project team with customer experience may not need the same level of domain and technology experience to produce the same level of performance as another team without customer experience.

Second, customer experience increases the ISD project team's communication efficiency and effectiveness, which may also help reduce the significance of domain and technology experience. In ISD, the customer typically provides the functional and technical requirements and constraints, which the ISD project team collects and interprets so that they may be transformed into an information system. Hence, it is imperative that the project team has the capability to communicate effectively with the customer to elicit the requirements and to understand them correctly (Davis 1982). Although we have already discussed the role of customer experience in providing information similar to what is provided by domain and technology experience, customer experience also provides additional benefits to the ISD project team by fostering trust between the vendor and the customer, which enables both parties to build effective and efficient communication channels (Ring and van de Ven 1994). These channels enhance the project team's capability to collect the customer's requirements accurately (Clark et al. 2013). We hypothesize the following:

Hypothesis 2A (H2A)**.** *The ISD project team's domain experience and technology experience interact negatively (substitutive interaction) such that the positive effect of technology experience on the project team's performance becomes weaker when the project team has greater domain experience.*

Hypothesis 2B (H2B)**.** *The ISD project team's domain experience and customer experience interact negatively (substitutive interaction) such that the positive effect of domain experience on the project team's performance becomes weaker when the project team has greater customer experience.*

Hypothesis 2C (H2C)**.** *The ISD project team's technology experience and customer experience interact negatively (substitutive interaction) such that the positive effect of technology experience on the project team's performance becomes weaker when the project team has greater customer experience.*

2.4. Moderating Effects of ISD Project Complexity

Complexity has been known as an important factor that influences the cognitive load and process of task performers in ISD (Wood 1986, Xia and Lee 2005). We characterize the ISD context as involving disparate tasks (different domain, technology, and customer) and teams (different project team members) across ISD projects. The specific structures of the *task* and the *team* would impact the level of complexity for the ISD project (Xia and Lee 2005, Langer et al. 2014). Task complexity is usually defined as the difficulty in transforming task inputs into outputs because of the interdependency or diversity among task elements (Espinosa et al. 2007, Xia and Lee 2005), whereas team complexity is generally defined as the difficulty in coordinating and communicating within a team because of the size of the team and the interdependency among team members (Xia and Lee 2005, Huckman et al. 2009). Both task and team complexities have been shown to hamper ISD project performance (Brooks 1995, Xia and Lee 2005, Wood 1986). Besides the above direct effects of the two types of ISD project complexity on ISD performance, the two types of complexity are expected to also affect the cognitive process of transfer of learning.

First, task complexity will hamper the learner's comprehension process by increasing the amount of information to be processed. As task complexity increases, it becomes more difficult for ISD project team members to comprehend the nature and essence of the current task and, consequently, more difficult for them to find connections between prior experience and the current task. This further decreases the likelihood that project team members will retrieve the relevant experience, resulting in greater difficulties in the transfer of past experience. ISD is essentially a knowledge-based task; therefore, task complexity is influenced by knowledge complexity. It is intuitive that a task dealing with more complex knowledge is more difficult to conduct than a task dealing with less complex knowledge. Given that each type of knowledge has its own learning effects, knowledge complexity is expected to affect the learning effect of its type. Therefore, we hypothesize the following:[4]

Hypothesis 3A (H3A)**.** *The ISD project team's technology experience and the ISD project's technology complexity interact negatively (offsetting interaction) such that the positive effect of technology experience on the project team's performance becomes weaker when the project has higher technology complexity.*

Hypothesis 3B (H3B)**.** *The ISD project team's customer experience and the ISD project's customer complexity interact negatively (offsetting interaction) such that the positive effect of customer experience on the project team's performance becomes weaker when the project has higher customer complexity.*

Second, ISD is a team task that relies on effective coordination among team members (Faraj and Sproull 2000). The complexity of the team decreases the performance of the learner's comprehension and coordination processes by increasing the amount of information to be processed. As team complexity increases, it becomes more difficult for the ISD project team to collectively retrieve relevant past experiences from individual team members and coordinate individuals' past experiences to be effectively transferred to the current task. This eventually leads to a decrease in the likelihood that team members will transfer their prior experiences to the current project. Hence, we hypothesize the following:

Hypothesis 4A (H4A)**.** *The ISD project team's domain experience and the project's team complexity interact negatively (offsetting interaction) such that the positive effect of domain experience on the project team's performance becomes weaker when the project has higher team complexity.*

Hypothesis 4B (H4B)**.** *The ISD project team's technology experience and the project's team complexity interact negatively (offsetting interaction) such that the positive effect of technology experience on the project team's performance becomes weaker when the project has higher team complexity.*

Hypothesis 4C (H4C)**.** *The ISD project team's customer experience and the project's team complexity interact negatively (offsetting interaction) such that the positive effect of customer experience on the project team's performance becomes weaker when the project has higher team complexity.*[5]

3. Research Methodology

3.1. Study Setting and Data Collection

Our research hypotheses are tested using archival data collected from a prominent global IT services company. The company, which acquired the Capability Maturity Model Integration (CMMI) Level 5 in 2004, provides contract-based custom ISD services to customers in a broad range of industries. Our data set contains detailed information on all ISD projects completed by the company dating back to the company's founding in the late 1980s, including (1) the domain, technology, and customer knowledge required by the projects; (2) the employees who worked on the projects; and (3) the employees' prior experiences in domain, technology, and customer knowledge. Our analysis sample consists of 497 ISD projects that ended between 2005 and 2007 and involved 2,393 unique employees. Most projects were large-scale development projects with an average project team size of approximately 9.2 members and an average duration of 10.1 months.

Different aspects of domain, technology, and customer knowledge are catalogued using a proprietary taxonomy that the company developed by combining professional standards used in industry such as the IEEE Guide to the Software Engineering Body of Knowledge (Bourque et al. 1999), IEEE Standards for Taxonomy for Software Engineering Standards (1002–1987) (Tripp and Fendrich 1987), IEEE Standards for Glossary of Software Engineering Terminology (610.12-1990) (Radatz et al. 1990), Gartner's classifications for information systems applications and markets, and standard industry classification, as well as inputs from internal and external experts. The elements at the most detailed level in the taxonomy, which we refer to as *knowledge categories* in this paper, are used to characterize the knowledge requirements for the ISD projects as well as to catalogue the experiences of the employees. Examples of domain knowledge categories include banking, insurance, e-government, military, e-commerce, human resources, manufacturing execution, product life cycle management, and health information systems. Examples of technology knowledge categories include specific programming languages (e.g., Java, C, HTML, COBOL, etc.), modeling techniques (e.g., Entity Relationship Diagram, Data Flow Diagram, Use Case Diagram, etc.), database systems (e.g., Oracle, Microsoft Access, DB2, etc.), operating systems (e.g., Windows, HP-Unix, etc.), and software development tools (e.g., Eclipse, Visual Studio, etc.). Customers are categorized simply by company names (e.g., Samsung Electronics, etc.).[6] As technologies, application domains, and customers' industries and names evolve over time, the company regularly updates its taxonomy. When the taxonomy is updated, the project knowledge requirement data and employees' experience records are also updated so that data integrity between the taxonomy and operational records is maintained.

3.2. Measures

3.2.1. Dependent Variable: Development Effort. Consistent with most prior learning studies in software engineering and information systems (e.g., Boh et al. 2007, Espinosa et al. 2007, Narayanan et al. 2009), we use development effort as the main dependent variable because it is objective and more appropriate for capturing learning effects compared to other measures of performance. We operationalize effort (*Effort*) as the actual labor hours incurred for project completion. For this measure, given that a project team that incurs less effort in completing the ISD project is deemed to have achieved better performance, a smaller number represents better performance.

Development effort has trade-offs with the quality of ISD projects in that less effort can be achieved at the detriment of low quality (e.g., finishing a project with less effort but many defects), and vice versa. Therefore, quality needs to be controlled for the effort measure to reflect performance more accurately (Espinosa et al. 2007). The company we study strictly controls the quality of ISD projects—every ISD project must pass an extensive and thorough final release test, conducted by an independent testing group, to be deemed completed. Therefore, our dependent variable is error free (i.e., without any delivery defect), similar to the ones used in Espinosa et al. (2007). It measures total development effort of an ISD project until the project delivers the promised system without any error to the customer. Similar to prior studies (Espinosa et al. 2007, Narayanan et al. 2009), our effort data are also skewed; therefore, we log it.

3.2.2. Independent Variables: Domain, Technology, and Customer Experience. Prior experience in the required domain ($Experience_{Dom}$), technology ($Experience_{Tech}$), and customer ($Experience_{Cust}$) knowledge categories is measured as the total number of prior projects in which the project team members had used those knowledge categories in domain, technology, and customer.[7] For instance, if a team member had previously used Java and developed a retail e-banking application in a project for Bank A, then the employee is said to have accumulated one unit of experience in retail e-banking (as domain knowledge), one unit of experience in Java (as technology knowledge), and one unit of experience in Bank A (as customer knowledge).[8]

To operationalize the experience at the project team level, we use the average across knowledge requirements, normalized by the number of developers in the project team.[9] More formally, the project team's cumulative experience for each ISD knowledge type is computed as $Experience_{Type} = \sum_{i=1}^{N}\sum_{j=1}^{M_{Type}} p(i,k_j)/(N \times M_{Type})$, where $Type = \{Dom, Tech, Cust\}$; N is the number of developers in the focal project team; M_{Type} is the number of knowledge categories required by the focal project for a given knowledge type; k_j represents the jth required knowledge category; and $p(i,k_j)$ represents the distinct number of prior projects in which member i had experienced knowledge category k_j.

3.2.3. Moderator Variables. We use the number of project team members for team complexity (*TeamComplexity*) and the number of distinct technology knowledge categories required by the project as a proxy for technology complexity ($TaskComplexity_{Tech}$) following the related literature (Xia and Lee 2005, Espinosa et al. 2007, Langer et al. 2014). For customer complexity ($TaskComplexity_{Cust}$), we use an indicator variable for whether (0) or not (1) the customer is within the same conglomerate company group that the IT services company belongs to. This company is part of a global conglomerate group that consists of various companies in many different industries. Although each company in the group is independent and operates in different industries and contexts, they share many commonalities, including standard communication systems, organizational culture, and business processes. Working with a customer within the conglomerate group is likely to be less complex than working with a customer not in the group because of shared communication systems, culture, and processes.

3.2.4. Control Variables. ISD project performance may be influenced by additional factors related to the team members, the project, the focal organization, and the external environment. Accordingly, our controls include general project experience (team member level); project size, development process type, customer industry, and team familiarity (project level); resource availability (organization level); and macroeconomic conditions (external environment). See Table 1 for details of these operationalizations.[10]

4. Results

Following the standard practice for analyzing models with interaction effects (Aiken et al. 1991), we estimate our ordinary least squares (OLS) linear regression models hierarchically (see Table 2)—a baseline model for development effort that includes only the control variables (Model 1); add the moderator variables (Model 2); add the domain, technology, and customer experience independent variables (Model 3); add all two-way interaction variables (Model 4); and finally, add the three-way interaction variable among domain, technology, and customer experiences (Model 5). Such a hierarchical approach allows us to check if the inclusion of these variables increases the explanatory power of the models. The results show that the moderator variables add significant explanation power to Model 1 (Model 1 versus Model 2; $\Delta R^2 = 0.0539$, $F = 10.93$, $p < 0.01$), implying that the moderator variables need to be used as controls in Model 3 as well. Domain, technology, and customer experiences significantly increase explanatory power (Model 2 versus Model 3: $\Delta R^2 = 0.0315$, $F = 16.90$, $p < 0.01$). The two-way interaction terms also significantly increase the explanation power of the model (Model 3 versus Model 4: $\Delta R^2 = 0.0666$, $F = 29.40$, $p < 0.01$). However, the three-way interaction term among domain, technology, and customer experiences do not add to the explanatory power (Model 4 versus Model 5: $\Delta R^2 = 0.0001$, $F = 0.09$, ns).

4.1. Main Results

The results for Model 3 show that domain and technology experiences exhibit significant learning effects, while customer experience does not ($Experience_{Dom}$: $\beta = -0.43$, $p < 0.01$; $Experience_{Tech}$: $\beta = -0.268$; $p < 0.01$; $Experience_{Cust}$: $\beta = 0.085$, ns). Thus, H1A and H1B are supported, but H1C is not. Model 4 is used to test Hypotheses 2–4. H2 posits that domain, technology, and customer experiences are substitutes for one another so that when ISD project teams have prior experience with one dimension, experience in either of the other two dimensions would be less beneficial to ISD projects compared to when the teams are not familiar with that dimension. H2A is supported but H2B and H2C are not ($Experience_{Dom} \times Experience_{Tech}$: $\beta = 0.031$, $p < 0.01$; $Experience_{Dom} \times Experience_{Cust}$: $\beta = 0.002$, ns; $Experience_{Tech} \times Experience_{Cust}$: $\beta = -0.056$, ns). H3 proposes that with more complex ISD projects in

Table 1. Operationalization of Control Variables

Level	Control	Description
Individual	General project experience	To tease out the effects of general project experience (i.e., learning by repeating ISD projects, without considering the knowledge requirement), we include a measure of the project team's prior project experience (*ProjectExperience*) as the average number of distinct prior projects the team members had participated in (irrespective of whether those projects required the knowledge categories that the focal project requires).[a]
Project	Project size	We use the project's estimated effort (*EstEffort*) and duration (*EstProjectDuration*) as proxies for project size (Huckman et al. 2009). We use estimates rather than actuals to avoid potential endogeneity problems (Huckman et al. 2009).
	Development process type	The type of development process may affect ISD project performance. We use a binary variable (*ProcessType*) to represent whether the project's development process is based on iterations (1) or traditional waterfall (0).
	Customer industry	We include industry sector dummies (*IndustrySector*) for public, manufacturing/production, and financial sectors, with the service sector as the base case.
	Team familiarity	Familiarity among team members is important for effective collaboration (Reagans et al. 2005, Boh et al. 2007, Huckman et al. 2009, Huckman and Staats 2011). Team familiarity breeds transactive memory within teams that aids collaboration through team members' knowledge about *who knows what* (Faraj and Sproull 2000, Wegner 1987). Following past research (e.g., Boh et al. 2007, Huckman et al. 2009, Reagans et al. 2005), we include the control variable *TeamFamiliarity*, computed as $TeamFamiliarity = \sum_{i=1}^{N-1}\sum_{l=i}^{N} 2w_{il}/(N(N-1))$, where w_{il} is the number of prior projects in which team members i and l had worked together and N is the number of team members.
Organization	Resource availability	The organization's situation in terms of resource availability may influence whether or not appropriate resources are available for assignment to projects, which is likely to have an impact on ISD project performance (Wiersma 2007). To account for this, we include a measure of overall workforce availability (*WorkforceAvailability*) using the ratio of developers not working in projects to the total number of developers in the company as a control. A lower workforce availability implies that fewer idle resources are available for assigning to new projects, resulting in a situation where finding appropriate team members to assign to projects is more constrained.
Environment	Macroeconomic conditions	We include year dummies for the project end year (Y_{2005} and Y_{2006}, with 2007 as the baseline), to control for macroeconomic conditions, such as inflation or business cycle and technological progress, that are independent of experience (Boh et al. 2007, Argote 2013).

[a]In our data set, all projects followed the company's standard project management processes (e.g., CMMI processes), and since all of them were for external clients, they required a certain level of interpersonal and communication skills. Hence, *ProjectExperience* may also control for the level of experience of the ISD knowledge that is not captured by domain, technology, and customer experience.

terms of the task, ISD project teams would have more difficulty in identifying and retrieving relevant prior experience so that the positive effects of technology (H3A) and customer (H3B) experiences on ISD project performance would diminish. Both H3A and H3B are supported ($Experience_{Tech} \times TaskComplexity_{Tech}$: $\beta = 0.009$, $p < 0.01$; $Experience_{Cust} \times TaskComplexity_{Cust}$: $\beta = 0.182$, $p < 0.1$). H4 posits that with more complex ISD projects in terms of the team, ISD project teams would have more difficulty in coordinating team members' transfer of prior experience so that the positive effects of team members' domain (H4A), technology (H4B), and customer (H4C) experiences on ISD project performance would diminish. All three hypotheses (H4A–H4C) are supported ($Experience_{Dom} \times TeamComplexity$: $\beta = 0.004$, $p < 0.05$; $Experience_{Tech} \times TeamComplexity$: $\beta = 0.115$, $p < 0.01$; $Experience_{Cust} \times TeamComplexity$: $\beta = 0.016$, $p < 0.1$).

4.2. Supplementary Analysis: Schedule Delay

The software engineering and information systems literature have consistently reiterated the challenges of assessing ISD project performance properly (e.g., Banker and Kemerer 1989) because ISD project performance has multiple dimensions and they may not always work in a synchronized manner (e.g., low schedule delay but high effort, etc.). It has been observed that development effort is the most appropriate measure to capture learning effects in software development (Boh et al. 2007, Espinosa et al. 2007, Narayanan et al. 2009); that is why it is used as the dependent variable in our main analysis in Section 4.1. That being said, other performance measures may still provide additional nuanced insights. Adherence to the planned schedule is one such alternative performance dimension and has been used along with effort in prior learning studies (Huckman et al. 2009, Huckman and Staats 2011). Therefore, as a supplementary analysis, we use schedule delay (*ScheduleDelay*), computed as the difference between the project's actual and planned end dates, as an additional dependent variable. With two dependent variables, since we cannot exclude the possibility that their error terms may be correlated, we use seemingly unrelated regression (SUR) models (Greene 2003). Table 3 shows the results.[11]

Table 2. Main Results—ln(*Effort*)

Variables	Model 1		Model 2		Model 3		Model 4		Model 5	
Intercept	2.556***	0.281	2.071***	0.280	2.333***	0.287	2.418***	0.258	2.405***	0.269
EstEffort	0.007***	0.001	0.003***	0.001	0.003***	0.001	0.004***	0.001	0.004***	0.001
EstProjectDuration	0.023***	0.008	0.030***	0.008	0.026***	0.007	0.030***	0.006	0.030***	0.006
ProjectExperience	−0.021***	0.007	−0.014**	0.006	0.004	0.007	0.009	0.006	0.009	0.006
TeamFamiliarity	−0.032**	0.016	−0.031**	0.015	−0.024	0.016	−0.026**	0.011	−0.025**	0.011
ProcessType	0.267**	0.117	0.137	0.102	0.120	0.087	0.101	0.063	0.102	0.063
WorkforceAvailability	−0.001	0.004	0.003	0.004	0.001	0.004	0.001	0.003	0.001	0.003
$IndustrySector_1$	0.515***	0.129	0.575***	0.121	0.539***	0.111	0.293***	0.100	0.292***	0.100
$IndustrySector_2$	0.485***	0.132	0.439***	0.109	0.402***	0.097	0.249***	0.090	0.249***	0.090
$IndustrySector_3$	0.270*	0.153	0.255**	0.128	0.283**	0.115	0.114	0.102	0.115	0.103
Y_{2005}	−0.101	0.085	−0.169*	0.090	−0.190**	0.096	−0.162**	0.077	−0.162**	0.078
Y_{2006}	0.059	0.088	−0.020	0.083	−0.013	0.081	−0.029	0.070	−0.029	0.070
TeamComplexity			0.023***	0.007	0.021***	0.007	−0.008	0.008	−0.008	0.008
$TaskComplexity_{Tech}$			0.006***	0.002	0.005***	0.002	0.003***	0.001	0.003***	0.001
$TaskComplexity_{Cust}$			−0.128	0.094	−0.036	0.090	−0.133	0.106	−0.134	0.107
$Experience_{Dom}$					−0.043***	0.012	−0.106***	0.023	−0.102***	0.028
$Experience_{Tech}$					−0.268***	0.065	−0.763***	0.129	−0.749***	0.154
$Experience_{Cust}$					0.085	0.086	−0.030	0.145	−0.013	0.156
$Experience_{Dom} \times Experience_{Tech}$							0.031***	0.011	0.028	0.018
$Experience_{Dom} \times Experience_{Cust}$							0.002	0.008	0.000	0.012
$Experience_{Tech} \times Experience_{Cust}$							−0.056	0.049	−0.072	0.077
$Experience_{Dom} \times TeamComplexity$							0.004**	0.002	0.004**	0.002
$Experience_{Tech} \times TeamComplexity$							0.115***	0.015	0.115***	0.015
$Experience_{Cust} \times TeamComplexity$							0.016*	0.009	0.015	0.009
$Experience_{Tech} \times TaskComplexity_{Tech}$							0.009***	0.003	0.009***	0.003
$Experience_{Cust} \times TaskComplexity_{Cust}$							0.182*	0.105	0.184***	0.106
$Experience_{Dom} \times Experience_{Tech} \times Experience_{Cust}$									0.002	0.008
N	497		497		497		497		497	
R^2	0.6310		0.6849		0.7164		0.7830		0.7831	
Adj R^2	0.6226		0.6760		0.7063		0.7715		0.7710	
ΔR^2			0.0539		0.0315		0.0666		0.0001	
$F(\Delta R^2)$			10.93***		16.90***		29.40***		0.09	

Note. Cluster robust standard errors with the customer as the primary sample unit are reported in the second column of each model (Moulton 1986).

*$p < 0.1$; **$p < 0.05$; ***$p < 0.01$.

The SUR models for development *Effort* produce the same results as the OLS models reported in Table 2. The results for *ScheduleDelay* show that domain, technology, and customer experiences exhibit significant learning effects ($Experience_{Dom}$: $\beta = -1.619$, $p < 0.1$; $Experience_{Tech}$: $\beta = -6.318$; $p < 0.1$; $Experience_{Cust}$: $\beta = -6.712$, $p < 0.05$ in Model 3). Hence, H1A–H1C are supported for schedule delay. The results in Model 4, which test Hypotheses 2–4, find substitutive interaction effects between domain and customer experiences ($Experience_{Dom} \times Experience_{Cust}$: $\beta = 1.574$, $p < 0.01$) and technology and customer experiences ($Experience_{Tech} \times Experience_{Cust}$: $\beta = 7.226$, $p < 0.1$). Therefore, H2B and H2C are supported. The interaction effect between domain and technology experiences is not substitutive ($Experience_{Dom} \times Experience_{Tech}$: $\beta = -1.714$, $p < 0.05$) in the schedule delay model. H3 and H4 are not supported in this model.

4.3. Discussion of the Results

Overall, all of our hypotheses are supported for either effort or schedule delay (or both). Among the 11 hypotheses, two (H1A and H1B) on the learning effects of domain and technology experiences are supported in both effort and schedule delay models. The moderating effects of task complexity (H3A and H3B) and team complexity (H4A, H4B, and H4C) and the substitutive interaction effect between domain and technology experiences (H2A) are supported in the effort models only. The learning effect of customer experience (H1C) and the substitutive interaction effects of customer experience with domain (H2B) and technology (H2C) experiences are supported in the schedule delay models only.

Interestingly, the hypotheses not supported in the effort models but supported in the schedule delay models (i.e., H1C, H2B, and H2C) are all related to customer experience. Schedule delay is determined by both project plans and actuals. ISD project teams may reduce schedule delay by creating a realistic (or accurate) project plan and/or by reducing the actual project duration. Given the difficulties of reducing the actual execution duration of an ISD project (Brooks 1995),

Table 3. Supplementary Analysis Results—ln(*Effort*) and *ScheduleDelay*

Variables	Model 1		Model 2		Model 3		Model 4		Model 5	
					ln(*Effort*)					
Intercept	2.556***	0.277	2.071***	0.276	2.333***	0.282	2.418***	0.252	2.405***	0.261
EstEffort	0.007***	0.001	0.003***	0.001	0.003***	0.001	0.004***	0.001	0.004***	0.001
EstProjectDuration	0.023***	0.008	0.030***	0.008	0.026***	0.007	0.030***	0.006	0.030***	0.006
ProjectExperience	−0.021***	0.007	−0.014**	0.006	0.004	0.007	0.009	0.006	0.009	0.006
TeamFamiliarity	−0.032**	0.015	−0.031**	0.015	−0.024	0.016	−0.026**	0.011	−0.025**	0.011
ProcessType	0.267**	0.116	0.137	0.100	0.120	0.085	0.101*	0.061	0.102*	0.061
WorkforceAvailability	−0.001	0.004	0.003	0.004	0.001	0.004	0.001	0.003	0.001	0.003
$IndustrySector_1$	0.515***	0.128	0.575***	0.120	0.539***	0.109	0.293***	0.098	0.292***	0.098
$IndustrySector_2$	0.485***	0.131	0.439***	0.107	0.402***	0.096	0.249***	0.088	0.249***	0.088
$IndustrySector_3$	0.270*	0.151	0.255**	0.126	0.283**	0.113	0.114	0.099	0.115	0.100
Y_{2005}	−0.101	0.084	−0.169*	0.089	−0.190**	0.095	−0.162**	0.075	−0.162**	0.075
Y_{2006}	0.059	0.087	−0.020	0.082	−0.013	0.080	−0.029	0.068	−0.029	0.068
TeamComplexity			0.023***	0.007	0.021***	0.006	−0.008	0.008	−0.008	0.008
$TaskComplexity_{Tech}$			0.006***	0.002	0.005***	0.001	0.003***	0.001	0.003***	0.001
$TaskComplexity_{Cust}$			−0.128	0.093	−0.036	0.089	−0.133	0.103	−0.134	0.104
$Experience_{Dom}$					−0.043***	0.011	−0.106***	0.022	−0.102***	0.027
$Experience_{Tech}$					−0.268***	0.064	−0.763***	0.126	−0.749***	0.149
$Experience_{Cust}$					0.085	0.085	−0.030	0.141	−0.013	0.151
$Experience_{Dom} \times Experience_{Tech}$							0.031***	0.011	0.028	0.017
$Experience_{Dom} \times Experience_{Cust}$							0.002	0.008	0.000	0.012
$Experience_{Tech} \times Experience_{Cust}$							−0.056	0.047	−0.072	0.075
$Experience_{Dom} \times TeamComplexity$							0.004**	0.002	0.004**	0.002
$Experience_{Tech} \times TeamComplexity$							0.115***	0.014	0.115***	0.014
$Experience_{Cust} \times TeamComplexity$							0.016*	0.009	0.015*	0.009
$Experience_{Tech} \times TaskComplexity_{Tech}$							0.009***	0.003	0.009***	0.003
$Experience_{Cust} \times TaskComplexity_{Cust}$							0.182*	0.103	0.184*	0.103
$Experience_{Dom} \times Experience_{Tech} \times Experience_{Cust}$									0.002	0.008
					ScheduleDelay					
Intercept	128.524***	47.676	139.286***	50.730	152.890***	52.694	162.629***	53.401	163.404***	54.001
EstEffort	0.069*	0.037	0.160*	0.093	0.167*	0.095	0.176*	0.093	0.176*	0.093
EstProjectDuration	−3.617**	1.664	−3.838**	1.715	−3.991**	1.732	−3.949**	1.621	−3.950**	1.622
ProjectExperience	0.052	0.379	−0.090	0.385	0.909*	0.486	0.637	0.530	0.649	0.533
TeamFamiliarity	−1.466*	0.816	−1.517*	0.792	−1.426*	0.761	−1.580*	0.874	−1.596*	0.887
ProcessType	−5.871	5.402	−3.579	4.868	−2.434	4.635	−3.564	5.328	−3.620	5.325
WorkforceAvailability	−1.331**	0.559	−1.452**	0.573	−1.529***	0.577	−1.535***	0.561	−1.535***	0.561
$IndustrySector_1$	6.480	6.492	−10.467	9.270	−10.375	9.308	−15.497	10.219	−15.485	10.216
$IndustrySector_2$	−3.390	5.172	−9.115	5.748	−11.939*	6.208	−16.335**	6.682	−16.351**	6.698
$IndustrySector_3$	26.944**	11.538	19.006*	10.515	16.583*	10.092	12.279	9.031	12.213	9.007
Y_{2005}	−23.335***	8.314	−26.875***	10.077	−29.206***	10.584	−31.778***	10.968	−31.812***	11.008
Y_{2006}	−10.100	8.895	−10.285	10.070	−10.885	10.015	−11.785	10.057	−11.819	10.082
TeamComplexity			−1.000	1.219	−1.096	1.227	−2.285	1.969	−2.300	1.979
$TaskComplexity_{Tech}$			−0.018	0.157	−0.046	0.148	0.003	0.204	0.003	0.204
$TaskComplexity_{Cust}$			21.278***	7.513	19.726***	7.387	21.726***	8.480	21.783***	8.488
$Experience_{Dom}$					−1.619*	0.951	−0.382	2.032	−0.589	2.194
$Experience_{Tech}$					−6.318*	3.430	−13.872	10.330	−14.694	11.349
$Experience_{Cust}$					−6.712**	3.387	−27.355**	10.826	−28.346**	12.239
$Experience_{Dom} \times Experience_{Tech}$							−1.714**	0.841	−1.543	1.049
$Experience_{Dom} \times Experience_{Cust}$							1.574***	0.603	1.738**	0.789
$Experience_{Tech} \times Experience_{Cust}$							7.226*	3.837	8.173	5.430
$Experience_{Dom} \times TeamComplexity$							−0.086	0.189	−0.083	0.189
$Experience_{Tech} \times TeamComplexity$							2.938	2.314	2.939	2.315
$Experience_{Cust} \times TeamComplexity$							1.332	1.140	1.350	1.156
$Experience_{Tech} \times TaskComplexity_{Tech}$							0.070	0.201	0.069	0.201
$Experience_{Cust} \times TaskComplexity_{Cust}$							−1.606	6.768	−1.678	6.735
$Experience_{Dom} \times Experience_{Tech} \times Experience_{Cust}$									−0.143	0.432
N	497		497		497		497		497	
Pseudolikelihood	−14,045.12		−18,143.31		−20,183.98		−29,354.49		−30,596.92	

Note. Cluster robust standard errors with the customer as the primary sample unit are reported in the second column of each model (Moulton 1986).

$^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$.

schedule delay is likely to be influenced by the accuracy of project plans. We conjecture that customer experience mainly influences how plans are developed rather than the actual effort incurred. The results of the learning effect of customer experience (H1C), which show that customer experience reduces schedule delay but does not affect effort, support this idea. Customer experience increases the ISD project team's general understanding on the customer's style, conditions, plans, and constraints, and also the project team's bargaining power with the customer. The increased understanding and bargaining power may enhance the ISD project team's ability to develop a *more realistic and accurate project plan*. This certainly overlaps with the benefits from domain and technology experience in enhancing the project planning ability of the ISD project team by helping it better understand the target application's domain and technical platforms. On the other hand, it seems relatively unlikely for customer experience to substitute for the benefit of domain and technical experience in reducing the actual effort spent to understand domain and technical issues, unless customer experience provides reusable knowledge related to the target application's domain and technologies. Customers may use a variety of technical environments for different systems, and much of the experience with the focal customer may have been in a different domain or technical setting. Prior experience working with the same customer, but in different domains and technical environments, may not prove to be very useful for substituting for the effects of prior experience in domain or technology in reducing the actual effort.

Another main difference between the results for effort and schedule delay is the moderating effects of ISD project complexity (Hypotheses 3 and 4). Similar to Espinosa et al. (2007), who find a negative interaction effect between task familiarity and task complexity, we find that the effect of the three types of experience is diminished as both types of ISD project complexity increase. However, interestingly, these moderating effects are only observed with respect to effort but not with respect to schedule delay. This is perhaps due to the difference between absolute and relative performance measures. Recall that effort is an absolute measure of ISD performance (the actual effort exerted), whereas schedule delay is a relative measure (the difference between planned and actual dates). ISD project complexity may influence both actuals and plans in similar ways, in which case relative measures may not be able to capture the change, whereas absolute measures may do so. For example, ISD project complexity may increase the actual duration (Xia and Lee 2005), but it may also help project teams, especially those with more experience, develop more realistic project plans, which would effectively accommodate expectations of the additional effort required and would mitigate the offsetting moderating effect of complexity. In this case, the absolute measure (actual effort or duration) would capture the moderating effect of ISD project complexity (H3 and H4), while the relative measure (planned minus actual duration) might not.[12] The interaction term between domain and technology experiences has mixed results for effort and schedule delay—a substitutive interaction effect is found for effort as expected (H2A), while a complimentary interaction effect is observed for schedule delay. These mixed results suggest that domain and technology experiences are substitutive in reducing *actual* effort; however, having both domain and technology experiences together has a synergetic positive effect on developing realistic project *plans*.

Now, we turn our attention to discussing the implications of our results. Regarding the results for the main effects of prior experience on ISD project performance (Model 3), as expected (Hypothesis 1), ISD project teams' domain and technology experiences show significant effects on reducing both the development effort and schedule delay, whereas customer experience exhibits a significant effect on reducing the schedule delay. This highlights the importance of domain, technology, and customer experience and that learning effects do occur for these three knowledge types in the ISD context where these three knowledge types are disparately assembled to deliver various kinds of information systems for different customers across ISD projects. This also shows that even when project teams are differently composed across ISD projects, the teams are able to use past experience for subsequent ISD tasks via the collective transfer of learning of team members. Our regression estimates (Model 3 in Table 3) show that, for a typical ISD project, if the project team had one more unit of domain (or technology) experience, the project could have reduced development effort by 4.2% (or 23.5%) and schedule delay by 1.6 (or 6.3) days, whereas with one more unit of customer experience, the project could have reduced schedule delay by 6.7 days.

With respect to the interaction effects among various experiences (Hypothesis 2), the results show that as domain or technology experience increases, the effect of customer experience diminishes. However, this effect only occurs for schedule delay but not for effort. Regarding the interaction between domain and technology experiences, as domain experience increases, the effect of technology experience on effort diminishes as expected but the positive effect on schedule delay amplifies. For a typical project, if the project team had one more unit of domain (or technology) experience, the beneficial effect (i.e., marginal decrease) in schedule delay would be reduced by 1.6 (or 7.2) days when customer experience is increased by one unit. Furthermore, with one more unit of technology experience, the marginal rate of decrease in effort by increasing one unit of domain experience would be reduced by 3%, whereas the marginal decreases in schedule delay would be increased by 1.7 days.

The effects of technology and customer experience in reducing development effort are shown to be offset by task complexity (Hypothesis 3). For a typical project, the marginal rate of decrease in development effort by increasing one unit of technology experience would be reduced by 2% if the project's technology complexity is increased by one unit. Similarly, the marginal rate of decrease in effort by increasing one unit of customer experience would be reduced by 21.7% when the customer is not in the same conglomerate group (i.e., high customer complexity) compared to when it is (i.e., low customer complexity). Regarding the moderating effects of ISD project team complexity, team complexity is found to offset the effects of domain, technology, and customer experience in reducing development effort (Hypothesis 4). For a typical project, the marginal rate of decrease in effort by increasing one unit of domain, technology, and customer experience would be reduced by 0.4%, 27%, and 1.9%, respectively, when the size of the development team is increased by one unit.

4.4. Robustness of the Results

Several additional analyses are conducted to establish the robustness of our results. To conserve space, we only summarize the results here; details are presented in the online appendix. First, although we control for project size using estimated project duration and effort, other measures such as the number of function points (Kemerer 1993) have been proposed as appropriate measures of ISD project size. We have controlled for function points and found that including function points in the analysis does not alter our main results. Second, we have included knowledge depreciation (or forgetting) effects in our analysis to see whether depreciation effects exist in ISD and to verify whether our main findings are robust to the inclusion of possible knowledge depreciation. The results do not show any significant depreciation effects. Third, it is the general policy of the company we study not to let project managers select specific team members. Rather, the human resources manager at the superseding level is responsible for staffing projects to balance organization-wide resource requirements. However, project managers may still exert influence in the selection of team members, in which case the results may be biased because of the project team being endogenously determined (e.g., the project manager may prefer to select better-performing members, while shunning poorer-performing candidates). We have checked for potential selection bias using the approach used in Huckman et al. (2009), but have not observed any indication of such bias.

5. Conclusion

5.1. Contributions

The key contribution of our study is a better understanding of how learning effects occur in ISD. Because of distinct characteristics of ISD (disparate tasks, teams, and level of project complexity), observing learning effects and understanding the underlying mechanism of learning have been a challenge. Drawing on the theory of transfer of learning, our study proposes a novel way of conceptualizing the same experience, using the three main types of ISD knowledge, namely, domain, technology, and customer knowledge, and shows that repeated experiences at the level of these three knowledge types result in significant learning effects.

Moreover, this study provides a better appreciation of how experiences in the three essential ISD knowledge types interact with one another such that when the experience in a given type is more (or less) abundant, ISD project team members can rely less (or more) on the experience in another type to achieve the same level of project performance. For example, when the ISD project team lacks prior experience with the customer, it may rely more on domain and technology experience to execute the project efficiently. Our results suggest that customer experience, although distinct from domain or technology experience, shares common elements with domain and technology such that the learning effects are substitutive. Finding interaction effects among different types of experiences contributes significantly to both ISD and general organizational learning literatures because it enables us to better understand under what conditions one type of experience complements or substitutes for another (Argote and Miron-Spektor 2011).

Our study also shows that ISD project complexity is inherently irreducible in that the beneficial effects of domain, technology, and customer experience exhibit diminishing returns for more complex ISD projects. This implies that inherently complex tasks make it more difficult for a project team to effectively trigger the necessary cognitive processes to retrieve and exploit its prior experience, suggesting that a greater amount of prior experience is needed to maintain the same level of ISD project performance for more complex ISD projects. Some capabilities, which may not be attainable from repeated task experiences such as "practical intelligence" for dealing with uncertainty, have been found to be more beneficial to more complex ISD projects (Langer et al. 2014). However, when it comes to task experience, the literature generally suggests that as ISD projects get more complex, the inherent complexity eventually becomes irreducible and cannot be compensated by developers' task experience (Brooks 1995), and that for complex ISD tasks, dramatic improvements are not possible through task experience alone (Espinosa et al. 2007). The findings in the organizational learning literature are consistent with this. For example, Argote et al. (1995) show in their experiment with origami tasks that performance on simpler tasks improves more strongly with task experience than on more complex tasks. This study provides a theoretical explanation for why such negative interaction effects occur from the perspective of transfer of learning.

This study offers important managerial implications as well. Our empirical findings provide insights into practical concerns in ISD such as how to assign team members to projects to increase performance, how to develop and retain expertise in ISD, and how to design career development paths for information systems professionals. For example, project staffing has generally been viewed as a problem of matching the requirements of the information system with team members' knowledge (i.e., prior experience). The normative staffing solution is to staff the project team with only those developers who have the knowledge relevant to the project at hand. However, in many cases, this is not always feasible because of limited human resources and/or constraints as a consequence of prior engagements of specialized team members in other projects. Our results suggest that, all else being equal, if the ISD project team is more familiar with a customer, domain and technology experience becomes less impactful, and if the project is more complex, any experience with the domain, technology, or customer also becomes less impactful.

5.2. Limitations and Future Research

Like most studies, this study is not without limitations, and its results should be interpreted with such limitations in mind. One limitation is due to the fact that the phenomenon of interest in this study is the effect of microlevel variables (e.g., experiences of individual group members) on a macrolevel variable (e.g., project team performance). To our knowledge, there is no statistical technique for a micro-macro situation like this. For instance, hierarchical linear modeling can only deal with macro-micro situations (i.e., macrolevel independent variables and microlevel dependent variables). Currently, the only available workaround for dealing with the micro-macro multilevel data is to aggregate the microlevel variables into macrolevel ones (Chan 1998, Klein et al. 1994). In this paper, we use the group average aggregation approach, which has been considered to be methodologically appropriate for additive composition models like ours (Chan 1998). Such an aggregation approach has also been widely used to capture the team-level learning effects in the ISD context (e.g., Boh et al. 2007, Espinosa et al. 2007, Huckman et al. 2009, Langer et al. 2014) as well as in other contexts such as surgical teams (e.g., Reagans et al. 2005).

Another limitation is that our analysis is at the knowledge type level, not at the individual knowledge category level. Grouping similar knowledge categories into meaningful higher-level types, such as domain, technology, and customer, has been adopted in many related studies (e.g., Langer et al. 2014, Bapna et al. 2013), and this approach has typically provided meaningful insights. However, the intrinsic level issues still exist. As in other similar studies (e.g., Bapna et al. 2013), our findings that hold at the level of knowledge type (i.e., domain, technology, and customer knowledge) may not necessarily hold for every specific knowledge category (e.g., e-banking, Java, etc.). Moreover, other types of knowledge and skills, such as project management knowledge (Henry et al. 2007), interpersonal and communication skills (Langer et al. 2014), and behavioral skills (Jiang et al. 1998), are also relevant for ISD projects. We, however, limit our focus in this study to the classification of three main types of knowledge used in ISD, namely, domain, technology, and customer knowledge. Nonetheless, investigating the effects of other possible types of knowledge would be a fruitful area for future research.

Consistent with the prior literature (e.g., Espinosa et al. 2007, Xia and Lee 2005), we consider team and task complexities as two salient dimensions of complexity in ISD. We acknowledge, however, that ISD project complexity may involve other dimensions (e.g., team's geographical dispersion may also represent team structural complexity). It would be worthwhile to investigate the moderating effects of other meaningful dimensions of ISD project complexity.

Last, our empirical data, including the taxonomy for domain, technology, and customer knowledge, is from a large IT services firm that provides ISD services. This focused research setting increases the internal validity of the results. On the other hand, these results may be affected by the idiosyncrasies of the research site. Future research in other settings would be useful to further generalize the results.

Acknowledgments

The authors thank the senior editor, associate editor, and the anonymous reviewers for their constructive comments and valuable suggestions on the paper.

Endnotes

[1] Although ISD and ISM are similar in the sense that both deal with information systems, they are different in many aspects. One of the main differences is that ISM repeatedly deals with *existing systems*, while ISD usually deals with *new systems*. In other words, ISM tasks typically entail working with the same system, whereas ISD tasks typically call for producing different systems and often for different customers. This difference is expected to alter the patterns of experience accumulation and, as a result, lead to differences in learning curve effects between ISD and ISM. Despite this expected difference, few studies have theorized and/or empirically investigated learning curve effects in the ISD context. What we currently know about learning curve effects in information systems is primarily from studies in the ISM context (e.g., Boh et al. 2007, Narayanan et al. 2009).

This paper fills a gap in the literature by examining learning effects specifically in ISD.

[2] ISD is generally executed by a project team. Although the theory of transfer of learning was originated to explain learning effects at the individual level, it can be extended to explain learning effects at the team level (Ghosh et al. 2014, Schilling et al. 2003, Lewis et al. 2005). As Schilling et al. mention "theories of individual cognition should inform theories of group cognition . . . because . . . some processes of group cognition may actually take place at the individual level" (2003, p. 43). There are two different mechanisms by which teams store and utilize past experience for current tasks (Argote 2013, Schilling et al. 2003). First, teams rely on individual team members' memory to store and retrieve past experience. In other words, team learning is a collective representation of individual team members' transfer of learning (Ghosh et al. 2014). Second, teams use team-level (or organization-level) memory to store and retrieve past experience. Examples of team-level memory include routines (Nelson and Winter 1982) and transactive memory that aids collaboration through team members' knowledge about who knows what (Faraj and Sproull 2000, Wegner 1987). These two mechanisms contribute to the learning effects observed at the team level, but a team may rely more on one mechanism than the other depending on environmental factors (Argote 2013, Lewis et al. 2005, Wegner 1987).

[3] Whether two different types of experience have a positive or negative interactive transfer effect depends on how they are related to each other. If one type helps the learner to identify, retrieve, or transfer the other type, it may increase the learner's absorptive capacity and create a positive interactive transfer (i.e., complementary interaction) (Cohen and Levinthal 1990). On the other hand, if the two types are in conflict with each other, interference may occur, creating a negative interactive transfer (i.e., offsetting interaction) (Underwood 1957, Armstrong and Hardgrave 2007). Likewise, if they are redundant such that they capture similar information, either type of experience, by itself, could be sufficient to improve performance, creating a negative interactive transfer (i.e., substitutive interaction) (Argote and Ingram 2000, Walsh and Ungson 1991). Alternatively, the two types of experience could be unrelated or orthogonal, in which case they are simply additive in their learning effects without any interactive transfer effect.

[4] Because of a lack of appropriate measure for domain complexity in our data set, we do not propose a hypothesis for the interaction between domain experience and domain complexity.

[5] Although the mechanisms for the interaction effects among domain, technology, and customer experiences (H2) and the moderating effects of ISD project complexity (H3 and H4) are different (substitutive versus offsetting), they both amount to negative interaction effects.

[6] An illustration on how knowledge categories are used to define the knowledge requirements of an ISD project and the experience of a team member related to this project is provided in the online appendix.

[7] An alternative measure for the amount of prior experience could be based on the total duration instead of the number of trials. While our data set permits us to identify which knowledge categories were used for each of the projects, it is not, however, fine-grained enough for us to ascertain the relative amount of effort exerted for each of the knowledge categories. In other words, while our data set may tell us that the C programming language was used on a particular project that lasted 18 months, we do not know the proportion of these 18 months in which C was actually used. Therefore, duration-based measures would suffer from measurement errors. Furthermore, the organizational learning literature seems to agree that trial-based measures capture learning effects more effectively than duration-based measures at the team or organization levels (see Argote 2013, Darr et al. 1995). The information systems literature generally uses the number of trials to present task experience (e.g., Narayanan et al. 2009, Boh et al. 2007, Espinosa et al. 2007).

[8] A project member's experience data captures only the knowledge categories actually worked on by the member. Even when a knowledge category was required for a project, if the project member did not actually do any work related to it, it was not counted as experience. The team members' actual experiences in the projects are initially self-reported but later verified by project and human resource managers. Because employees' performance evaluations are based on actual experiences with the knowledge requirements, project team members are required to report all actual experiences correctly, and project and human resources managers are also required to carefully verify and corroborate them. The knowledge requirements for each project are initially defined during project planning by project managers based on the customer's needs and later revised during project execution, if necessary.

[9] The average experience across team members has been widely used to represent the team-level experience in the literature, including learning studies in information systems (Boh et al. 2007, Huckman et al. 2009, Langer et al. 2014, Espinosa et al. 2007). The underlying rationale for this average is to assume that every team member's experience equally contributes to the team. This assumption seems to be the most reasonable, given that the contribution of each member is varying rather than fixed across teams and that the average generally treats this variation reasonably well. The same rationale is used to deal with the unknown and varying contribution of experience in different knowledge categories. We sum all experiences for required knowledge categories and normalize it by the number of knowledge categories. For example, if a project requires Java and HTML (two technology categories) and a project member had previously worked on Java in two projects and HTML in one project, then the project member is said to have $(2+1)/2 = 1.5$ experiences in technology knowledge required for the project. The same assumption and operationalization has been used in related studies (e.g., Langer et al. 2014).

[10] The descriptive statistics for all variables at the project and team member levels and the intercorrelations among the variables are provided in the online appendix.

[11] When the regressors are the same across regression models with different dependent variables (as in our case), OLS produces unbiased estimators that are exactly the same as those obtained from SUR, although standard errors may differ (Greene 2003). To check if there are any differences in the hypothesis testing results, we have also run OLS for effort and schedule delay separately and found the results to be consistent.

[12] That being said, the actual schedule is not appropriate as a measure of efficiency since schedule is usually constrained by external factors such as the customer's internal plan. The more important measure to both the customer and the service provider is the extent to which the project meets the planned schedule, as captured by schedule delay.

References

Adelson B, Soloway E (1985) The role of domain experience in software design. *IEEE Trans. Software Engrg.* 11(11):1351–1360.

Aiken LS, West SG, Reno RR (1991) *Multiple Regression: Testing and Interpreting Interactions* (Sage, Newbury Park, CA).

Argote L (2013) *Organizational Learning: Creating, Retaining, and Transferring Knowledge*, 2nd ed. (Springer, New York).

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

Argote L, Miron-Spektor E (2011) Organizational learning: From experience to knowledge. *Organ. Sci.* 22(5):1123–1137.

Argote L, Insko CA, Yovetich N, Romero AA (1995) Group learning curves: The effects of turnover and task complexity on group performance. *J. Appl. Soc. Psych.* 25(6):512–529.

Armstrong DJ, Hardgrave BC (2007) Understanding mindshift learning: The transition to object-oriented development. *MIS Quart.* 31(3):453–474.

Banker RD, Kemerer CF (1989) Scale economics in new software development. *IEEE Trans. Software Engrg.* 15(10):1199–1205.

Bapna R, Langer N, Mehra A, Gopal R, Gupta A (2013) Human capital investments and employee performance: An analysis of IT services industry. *Management Sci.* 59(3):641–658.

Boh WF, Slaughter SA, Espinosa JA (2007) Learning from experience in software development: A multilevel analysis. *Management Sci.* 53(8):1315–1331.

Bourque P, Dupuis R, Abran A, Moore JW, Tripp L (1999) The guide to the software engineering body of knowledge. *IEEE Software* 16(6):35–44.

Brooks FP (1995) *The Mythical Man-Month: Essays on Software Engineering*, Anniversary ed. (Prentice-Hall, Upper Saddle River, NJ).

Chan D (1998) Functional relations among constructs in the same content domain at different levels of analysis: A typology of composition models. *J. Appl. Psych.* 83(2):234–246.

Clark J, Huckman RS, Staats BR (2013) Learning from customers: Individual and organizational effects in radiological services. *Organ. Sci.* 24(5):1539–1557.

Cohen WM, Levinthal DA (1990) Absorptive capacity: A new perspective on learning and innovation. *Admin. Sci. Quart.* 35(1):128–152.

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

Davis GB (1982) Strategies for information requirements determination. *IBM Systems J.* 21(1):4–30.

Day SB, Goldstone RL (2012) The import of knowledge export: Connecting findings and theories of transfer of learning. *Educational Psychologist* 47(3):153–176.

El Emam K, Koru AG (2008) A replicated survey of IT software project failures. *IEEE Software* 25(5):84–90.

Ellis HC (1965) *The Transfer of Learning* (MacMillan, New York).

Espinosa JA, Slaughter SA, Kraut RE, Herbsleb JD (2007) Familiarity, complexity, and team performance in geographically distributed software development. *Organ. Sci.* 18(4):613–630.

Ethiraj SK, Kale P, Krishnan MS, Singh JV (2005) Where do capabilities come from and how do they matter? A study in the software services industry. *Strategic Management J.* 26(1):25–45.

Ewusi-Mensah K (2003) *Software Development Failures* (MIT Press, Cambridge, MA).

Faraj S, Sproull LS (2000) Coordinating expertise in software development teams. *Management Sci.* 46(12):1554–1568.

Gagné RM (1965) *The Conditions of Learning* (Holt, Rinehart, and Winston, New York).

Garud R (1997) On the distinction between know-how, know-why, and know-what. Huff A, Walsh J, eds. *Advances in Strategic Management* (JAI Press, Greenwich, CT), 81–101.

Ghosh A, Martin X, Pennings JM, Wezel FC (2014) Ambition is nothing without focus: Compensating for negative transfer of experience in R&D. *Organ. Sci.* 25(2):572–590.

Gopal A, Gosain S (2010) The role of formal controls and boundary spanning in software development outsourcing: Implications for project performance. *Inform. Systems Res.* 21(4):960–982.

Greene WH (2003) *Econometric Analysis* (Prentice Hall, Englewood Cliffs, NJ).

Henry RM, McCray GE, Purvis RL, Roberts TL (2007) Exploiting organizational knowledge in developing IS project cost and schedule estimates: An empirical study. *Inform. Management* 44(6):598–612.

Huckman RS, Staats BR (2011) Fluid tasks and fluid teams: The impact of diversity in experience and team familiarity on team performance. *Manufacturing Service Oper. Management* 13(3): 310–328.

Huckman RS, Staats BR, Upton DM (2009) Team familiarity, role experience, and performance: Evidence from Indian software services. *Management Sci.* 55(1):85–100.

Jiang JJ, Klein G, Margulis G (1998) Important behavioral skills for IS project managers: The judgments of experienced IS professionals. *Project Management J.* 29(10):39–44.

Kantor PB, Zangwill WI (1991) Theoretical foundation for a learning rate budget. *Management Sci.* 37(3):315–330.

Kemerer CF (1993) Reliability of function points measurement: A field experiment. *Comm. ACM* 36(2):85–97.

Khatri V, Vessey I, Ramesh V, Clay P, Park S (2006) Understanding conceptual schemas: Exploring the role of application and IS domain knowledge. *Inform. Systems Res.* 17(1):81–99.

Klein KJ, Dansereau F, Hall RJ (1994) Levels issues in theory development, data collection, and analysis. *Acad. Management Rev.* 19(2):195–229.

Langer N, Slaughter SA, Mukhopadhyay T (2014) Project managers' practical intelligence and project performance in software offshore outsourcing: A field study. *Inform. Systems Res.* 25(2): 364–384.

Lee DMS, Trauth EM, Farwell D (1995) Critical skills and knowledge requirements of IS professionals: A joint academic-industry investigation. *MIS Quart.* 19(3):313–340.

Levinthal DA, March JG (1993) The myopia of learning. *Strategic Management J.* 14(S2):95–112.

Lewis K, Lange D, Gillis L (2005) Transactive memory systems, learning, and learning transfer. *Organ. Sci.* 16(6):581–598.

Moulton BR (1986) Random group effects and the precision of regression estimates. *J. Econometrics* 33(3):385–397.

Narayanan S, Balasubramanian S, Swaminathan J (2009) A matter of balance: Specialization, task variety, and individual learning in a software maintenance environment. *Management Sci.* 55(11):1861–1876.

Nelson RR, Winter SG (1982) *An Evolutionary Theory of Economic Change* (Belknap Press, Cambridge, MA).

Perkins DN, Salomon G (1994) Transfer of learning. Husen T, Postelwhite TN, eds. *International Handbook of Educational Research* (Oxford University Press, New York), 6452–6457.

Radatz J, Geraci A, Katki F (1990) *IEEE Standard Glossary of Software Engineering Terminology* (IEEE, New York).

Reagans R, Argote L, Brooks D (2005) Individual experience and experience working together: Predicting learning rates from knowing who knows what and knowing how to work together. *Management Sci.* 51(6):869–881.

Ring PS, van de Ven AH (1994) Developmental processes of cooperative interorganizational relationships. *Acad. Management Rev.* 19(1):90–118.

Royer JM (1979) Theories of the transfer of learning. *Educational Psychologist* 14(1):53–69.

Rubenstein D (2007) Standish group report: There's less development chaos today. *Software Development Times; Oyster Bay* 169:1, 19.

Schilling MA, Vidal P, Ployhart RE, Marangoni A (2003) Learning by doing something else: Variation, relatedness, and the learning curve. *Management Sci.* 49(1):39–56.

Schunk DH (2015) *Learning Theories: An Educational Perspective*, 7th ed. (Pearson, Boston).

Thorndike EL, Woodworth RS (1901) The influence of improvement in one mental function upon the efficiency of other functions. *Psych. Rev.* 8(1):247–261.

Tripp LL, Fendrich JW (1987) Taxonomy of software engineering standards: A development history. *Comput. Standards Interfaces* 6(2):195–205.

Underwood BJ (1957) Interference and forgetting. *Psych. Rev.* 64(1): 49–60.

Walsh JP, Ungson GR (1991) Organizational memory. *Acad. Management Rev.* 16(1):57–91.

Wegner DM (1987) Transactive memory: A contemporary analysis of the group mind. Mullen B, Goethals GR, eds. *Theories of Group Behavior* (Springer-Verlag, New York), 185–208.

Wiersma E (2007) Conditions that shape the learning curve: Factors that increase the ability and opportunity to learn. *Management Sci.* 53(12):1903–1915.

Wood RE (1986) Task complexity: Definition of the construct. *Organ. Behav. Human Decision Processes* 37(1):60–82.

Xia W, Lee G (2005) Complexity of information systems development projects: Conceptualization and measurement development. *J. Management Inform. Systems* 22(1):45–83.