BALANCING RIGOR, STANDARDIZATION, AND AGILITY IN DISTRIBUTED IS DEVELOPMENT PROCESS: AN AMBIDEXTERITY PERSPECTIVE

Gwanhoo Lee

glee@american.edu

Phone: (202) 885-1991

Fax: (202) 885-1992

J. Alberto Espinosa

alberto@american.edu

William DeLone

wdelone@american.edu

Kogod School of Business

American University

4400 Massachusetts Avenue, NW

Washington, DC 20016-8044

BALANCING RIGOR, STANDARDIZATION, AND AGILITY IN DISTRIBUTED IS DEVELOPMENT PROCESS: AN AMBIDEXTERITY PERSPECTIVE

ABSTRACT

Distributed information systems (IS) development faces daunting challenges including communication and coordination difficulties, increased user requirement uncertainty, and greater task complexity. To cope with such challenges, distributed IS teams attempt to build effective development process capabilities such as process rigor, process standardization, and process agility. However, the complex effects of these process capabilities on distributed IS development performance are not well understood or empirically validated. To fill this gap in our knowledge, we investigate how rigor, standardization, and agility of development process respectively affect the performance of the system delivered by a distributed team. Furthermore and more importantly, we investigate the notion of IS development process ambidexterity which is defined as the simultaneous presence of alignment and agility in development process, where rigor and standardization represent two dimensions of alignment. We examine if such process ambidexterity demonstrates a positive effect on system performance. We used hierarchical regression to analyze field data from project managers of distributed IS development. Our results support a positive main effect of rigor, standardization, and agility on system performance in distributed development. We find a positive interaction effect of rigor and agility, indicating a synergistic effect of process ambidexterity. Contrary to our expectation, however, we find a negative interaction effect of standardization and agility, indicating an offsetting effect of process ambidexterity. We discuss the theoretical and practical implications of these findings for balancing rigor, standardization, and agility in distributed development in order to achieve better system performance.

***Keywords*:** Process rigor; process standardization; process agility; process ambidexterity; process alignment; distributed information systems development; distributed team; system performance

BALANCING RIGOR, STANDARDIZATION, AND AGILITY IN DISTRIBUTED IS DEVELOPMENT PROCESS: AN AMBIDEXTERITY PERSPECTIVE

INTRODUCTION

Information systems (IS) these days are often developed across multiple locations, partly due to increased globalization, outsourcing, and offshoring [29]. A recent Gartner report [43] indicates that more than 90% of Fortune 500 organizations use some type of external resources for IT services delivery and that organizations typically spent 31 percent of IT spending on external IT services in 2010. Geographically distributed development has become not only feasible but also attractive and compelling, thanks to the widespread availability of specialized talent, lower cost labor, and technologies bridging geographic distance [14]. As a result, distributed development has become one of the major current trends in IS development [3].

As an IBM report [32] stated, "Going global offers many benefits. However, distributed organizations face more challenges than collocated teams," distributed IS teams face special challenges [8, 26]. Geographic dispersion causes increased complexity and uncertainty in team collaboration and places new demands on the development process, making it difficult for IS development to succeed [3, 38]. It leads to reduction in communication frequency [22], diminished likelihood of collaboration [45], less knowledge sharing [36], and increased costs [8]. As a result, the growing adoption of distributed development may exacerbate the historically-low success rate of IS development [67].

To cope with the challenges related to geographic dispersion , distributed IS teams might need strong and effective development process capabilities [48, 62] as the effectiveness of development processes has long been considered to be important for IS development success [65]. Due in part to increased complexity and uncertainty in the development environment, distributed IS teams appear to face two seemingly paradoxical demands for development process capabilities: process alignment and process agility [13, 63]. These paradoxical needs in distributed development are not well understood despite their apparent importance and, to the best of our knowledge, no prior research has investigated this phenomenon, which is a gap our research attempts to fill. Based on prior research [19, 33, 50, 52], we define *process alignment* as "the coherence among all activities in IS development process" and *process agility* as "the process capability to sense and respond to changing system requirements." On the one hand, distributed teams may need a high level of process alignment to carry out tasks with stability, consistency, accuracy, and efficiency to overcome difficulties in communication and coordination across locations [48]. On the other hand, distributed teams may also need a high level of process agility to deliver systems that meet changed user requirements resulting from greater uncertainty and dynamism in distributed environment [3]. Although any IS development projects may face this tension, its frequency and intensity are more pronounced in distributed development [63].

This research focuses on process rigor and process standardization as two important dimensions of process alignment. Consistent with prior literature [11, 13], we define *process rigor* as "the process capability that increases clarity, accuracy, and formality in the development process mainly through detailed planning and documentation." While process rigor may be important for virtually all IS development [60] and not uniquely important for distributed IS development, process standardization, as defined in this research, is especially relevant to the distributed development context. Consistent with prior literature [57, 58], we define *process standardization* as "the uniformity and consistency of IS development methods, techniques, and practices across development sites."

Although process capabilities such as rigor, standardization, and agility seem generally beneficial for IS development, the main effects of these process capabilities in the distributed development context have not been well understood or empirically validated. Therefore, empirical testing of these effects is needed to validate the value of such process capabilities in geographically dispersed development. While it is important to understand how process capabilities individually affect system performance in distributed development, a much more important question is whether or not the simultaneous presence of process alignment and process agility, which we term *IS development process ambidexterity*, has a positive effect on system performance above and beyond their main effect. Consistent with prior literature [68], we define IS development process ambidexterity as the process capability to simultaneously exhibit alignment and agility in IS development.

The organizational ambidexterity literature has shown that, although the tensions between alignment and adaptability exist, their simultaneous presence are positively correlated to higher firm performance [12]. Conceptually, this ambidexterity perspective can be applied to distributed IS development. Distributed IS teams often attempt to pursue not only process alignment but also process agility and face decisions how to best balance them. Therefore, it is critical for them to understand how different process capabilities interact to affect system performance. Although some recent studies have investigated the phenomena related to IS development ambidexterity, most of them tend to focus on understanding its antecedents [37, 63, 68]. To the best of our knowledge, no prior research has investigated the effect of IS development process ambidexterity on system performance. Consequently, IS teams are not well informed whether they should increase both alignment and agility simultaneously or make a tradeoff between the two. In sum, to fill this gap in our knowledge, this research aims to address the following question: *what are the main and interaction effects of process alignment (i.e., rigor and standardization) and agility on system performance in distributed IS development?*

In this research, we decompose process alignment into process rigor and process standardization, which represent two important capabilities of IS development process. One can think of process rigor as vertical alignment that helps align individual tasks from the beginning to the end of the project lifecycle and process standardization as lateral alignment that ensures individual tasks to be aligned between different locations. By examining process ambidexterity at this granular level, this research intends to discover more nuanced effects of process ambidexterity on system performance in distributed development.

The remainder of the paper is organized as follows. In the next section, we discuss theoretical background for this research. We then develop our hypotheses based on prior literature on IS development, distributed teams, and organizational ambidexterity. We also used our preliminary field interview data from IS project managers to support some of the hypotheses. We then discuss our research methods and test the hypotheses using our survey data and hierarchical regression. We present the results and conclude this paper with the discussion on the theoretical and practical implications of the findings for balancing rigor, standardization, and agility in distributed IS development.

THEORETICAL BACKGROUND

Process Ambidexterity in Distributed IS Development

In organizational research, the term *ambidexterity* has been used to refer to the organizational capability to do two seemingly different or even contradictory things simultaneously [61]. More specifically, organizational ambidexterity has been defined as an organization's ability to concurrently demonstrate alignment and adaptability [33], exploitation and exploration [35, 41], efficiency and flexibility [1] or incremental change and revolutionary change [69]. This research focuses on ambidexterity in terms of simultaneous presence of alignment and agility as these two dimensions are closely tied to important IS development process capabilities.

Increasingly more researchers have begun to shift their focus from trade-off to ambidexterity perspective to understand paradoxical synergistic effects of seemingly contradictory things [51]. The organizational ambidexterity literature has argued for the possibility of achieving both alignment and adaptability and its positive impact on organizational performance [33, 41, 69]. It has been argued that, although the tensions between alignment and adaptability exist, the most successful organizations reconcile them to a large degree, and in doing so enhance their long-term competitiveness and performance [12].

The ambidexterity perspective has been applied not only at the organizational level but also at the level of business unit and team [61]. In this research, we apply it to geographically dispersed IS development processes. An IS development process refers to the set of activities that are performed during the development of an information system [24]. These activities involve interactions among people, technology, methods, and procedures [39].

Process rigor and process standardization are identified as two key dimensions of process alignment. Rigorous and standardized development processes promise improved quality, maintainability, and efficiency [8]. The importance of process rigor has long been recognized even before distributed development became pervasive [60]. On the other hand, process standardization is a dimension of process alignment that is unique to distributed IS development as it refers to the uniformity and consistency of development process across different locations. With these two types of process alignment, this research examines two types of IS development process ambidexterity: process ambidexterity of rigor and agility and process ambidexterity of standardization and agility.

IS development may require both alignment and agility as the former ensures short-term effectiveness, whereas the latter fosters long-term effectiveness [6, 13, 60]. The notion of ambidexterity balancing alignment and agility in IS development has recently gained attention from IS researchers [3, 13, 37, 48, 68, 72]. One study, based on qualitative field interview data, has identified ambidextrous coping strategies that distributed IS teams used [48]. Other studies identified antecedents for IS development process ambidexterity [63] and investigated how to balance agility with control in IS development [37]. However, there is an important gap in the literature in terms of how process alignment (rigor and standardization) and process agility interact to affect distributed development performance.

Process Rigor, Standardization, and Agility

Process rigor has been long viewed as an important process capability and organizations have devoted their efforts to improving it [60]. It is characterized by clear definitions of roles, activities, work products, methods, and measures, detailed top-down planning, detailed documentation, and use of formal methods [6, 11, 42, 48]. It is especially emphasized by plan-based, structured approaches to IS development [13] and is facilitated by such methods as the Capability Maturity Model Integrated (CMMI) [4, 46, 60]. The structured, plan-driven development methodologies were widely implemented to reduce defects, improve quality and productivity, and increase user satisfaction. Process rigor is not easily achieved in practice as one study shows that only 6% of software developers rigorously adhere to pre-defined methods [31].

Although prior literature tends to suggest the general benefit of process rigor in IS development [42], little research has investigated its effect on system performance in distributed development. So, the positive effect of process rigor on system performance should not be taken for granted in the context of distributed development and needs empirical testing. Rigor seems to be important in distributed IS development environments in part because members of a distributed team do not interact as frequently as members of a collocated team, so it is more difficult to monitor and control tasks [38]. Rigor helps reduce ambiguity and increase clarity and accountability in IS development work. It also allows team members to understand how individual tasks fit into the entire project throughout the development lifecycle.

Process standardization has been considered another important process capability to improve IS development performance [57]. Most prior research has investigated it in terms of standardization across different projects rather than standardization across different locations within the same project [25, 57]. In one prior study, however, process standardization was defined as the consistent use of methodologies, tools, techniques, templates, and work practices across development sites [58]. In sum, little research has investigated the effect of process standardization on system performance in distributed development. While process rigor focuses on clarity, accuracy, formality, and reliability of IS development processes, process standardization mainly focuses on uniformity and consistency of IS development processes across development sites.

Distributed development decreases communication quality and frequency and increases the possibilities for conflict, misunderstanding, and breakdowns in communication [16]. It is very challenging for distributed teams to develop shared understanding because members of such teams do not stand on ‘common ground’[21]. A case study of a large offshore IT service provider suggests that the standardization of software development methodologies and templates across the remote sites facilitates knowledge transfer in geographically distributed teams by overcoming differences in work routines and methodologies derived from the diverse local contexts [58]. Standardized processes enable team members to set shared expectations, better understand each others' work, coordinate their activities more easily, and thus undertake system integration more effectively [48, 57]. Furthermore, standardized processes can reduce the negative impact of personnel turnover, lead to a cohesive organizational culture, and help overcome differences derived from diverse local contexts [40, 57, 58].

Since geographic dispersion tends to increase uncertainty and dynamism in the task environment, distributed teams might need flexibility and adaptability to be effective [64]. Process agility refers to the process capability to sense and respond to changing system requirements [19, 49, 50, 52]. In the late 1990s and the early 21st century, recognizing the detrimental consequences of the lack of process adaptability, organizations began to adopt agile development approaches such as XP (Extreme Programming) and Scrum to improve IS development agility in response to constantly changing requirements [9, 18]. The rapid adoption of agile development methods suggests that process agility is perceived by organizations to be an important capability for IS development. Agile development advocates tend to value agility over rigor and standardization, and argue that overly rigorous and standardized processes do not necessarily pay off and may even do harm [9]. The benefits of process agility have been demonstrated mostly for collocated and/or small-scale and development and generalizability of such findings to distributed development has not yet been validated [19, 26, 66]. As a result, the role of process agility in distributed development is poorly understood [2, 3].

Process agility enables IS teams to strategize their responses to changing system requirements and implement necessary changes in a timely and cost-effective manner [9, 17]. It lowers the cost of incorporating changes [7]. Due to inherent uncertainty in business and technology environments, changes in system requirements are inevitable [47, 49]. Process agility is needed especially when required changes are not anticipated or adequately specified at the onset of the project [71].

HYPOTHESES DEVELOPMENT

Drawing upon the literature on IS development, distributed teams, and organization ambidexterity as well as our field interview data, we posit in this study that process rigor, process standardization, and process agility respectively have a positive effect on system performance because geographically dispersed development environment requires such process capabilities. We also posit that rigor and agility and standardization and agility demonstrate a positive interaction effect on system performance as these process capabilities would benefit from each other. Figure 1 depicts our research model and hypotheses. In the following sections, we discuss our hypotheses in more detail.

Figure 1 about here

Main Effects of Process Rigor, Standardization, and Agility on System Performance in Distributed IS Development

Process rigor ensures clarity, formality, and detailed documentation and planning when a software team develops software. It helps align development tasks throughout the entire IS development lifecycle as each task is well defined and planned in advance and the relationships between different tasks are clearly communicated and documented [4]. It allows team members to develop the sense of control, predictability, and reliability in an otherwise uncertain and unstable task environment [63]. When development processes are not rigorous, distributed teams are more vulnerable to making errors and mistakes, resulting in costly rework. Process rigor enables team members to understand how their individual task activities contribute to the activities of others and the project as a whole, and also helps develop accurate expectations about the task activities that other members need to carry out. Therefore, lack of process rigor also may cause misalignment between tasks and thus create integration problems, leading to IS development failures [60]. One global project manager in our field interview stated, *"I don't want to make it too simplistic, but good communication, strong project management and clearly defined processes are the keys to make projects successful from my standpoint."* We argue that process rigor increases system performance because it enables distributed software teams to reduce errors and mistakes and to align and integrate tasks effectively.

H1. Process rigor positively affects system performance in distributed IS development.

While communication is a critical success factor for virtually all IS development work, communicating spontaneously, frequently, and unambiguously is not easy in geographically distributed environments [62]. Standardized processes across project locations help overcome communication barriers and facilitate coordination among team members distributed geographically who, otherwise, might have very different ways of carrying out tasks [58]. With standardized processes, team members can understand each other's work easily even if they are geographically dispersed. Standardized processes can save a great deal of time and effort because, without them, distributed teams would need to spend a significant amount of time and effort to continuously negotiate ground rules and protocols for task coordination. Since the benefits of co-presence are absent in geographically dispersed environments and given the communication challenges imposed by distance, process standardization can help team members know how they are expected to carry out their tasks and how others will carry out theirs.

Process standardization enables distributed IS teams to develop a shared mental model of how work is done [44, 55] and leads to a cohesive organizational culture through the common technical language, procedures, and goals [40]. One project manager in our field interview stated, *"One of the key things we have done right is, we have internally developed project management processes and tools that we use globally from wherever we are. We also have an extranet with our offshore partner and they use the same hardware and software that we use."* We argue that standardized IS development processes lead to higher system performance by reducing process variance across locations. Reduced process variance in turn reduces time-consuming coordination and negotiation to resolve differences and conflicts across locations.

H2. Process standardization across locations positively affects system performance in distributed IS development.

Change in user requirements during IS development is the rule rather than exception [49]. A distributed development environment exacerbates the uncertainty of system requirements in part because of difficulties in defining initial requirements. One study shows that 50% to 60% of the requirements changed during distributed development [63]. Furthermore, requirements in distributed development tend to be more volatile due to competitive environment changes occurring across multiple team boundaries. Thus, system performance in the distributed environment can be significantly undermined when IS development processes are not designed to effectively manage requirement changes. Agility is increasingly being seen as an essential capability for distributed development [66]. However, spatial and temporal boundaries in distributed teams tends to reduce the ability to respond to change [23]. One project manager in our preliminary field interview stressed that the lack of their vendor's process agility caused a tension as it had a high impact on project performance. "*Every time we ask for new deliverables* *or changes, we're always getting a lot of pushback* (*from the vendor*) *that they cannot meet our delivery dates.*"

We argue that process agility is particularly more important for geographically distributed IS development because communication is less effective in distributed environments, making it difficult and costly for the team to respond to changes. When development teams are collocated, they can coordinate their response to a requirement change by informal communication and immediate feedback [54, 70]. Conversely, geographically dispersed team members cannot rely as much on coordination by such informal communication and immediate feedback. Thus, they need agility built into development processes in order to adapt quickly with little informal communication. For example, short time-boxed development cycles can be useful in geographically dispersed environments because it fosters granular, structured and frequent handoffs and integration of work from one site to the next, improving the flow of information and thus the team's ability to respond to change [15]. We argue that a streamlined, efficient process that enables the IS team to sense critical requirement changes and to strategize and implement appropriate responses increases system performance in geographically distributed development.

H3. Process agility positively affects system performance in distributed IS development.

Interaction Effects of Process Alignment and Process Agility on System Performance in Distributed IS Development

Derived from the literature in organizational ambidexterity, organizational learning, and strategic management, we hypothesize how the simultaneous presence of process alignment and process agility affects system performance in distributed development. Consistent with the organizational ambidexterity literature [12], we propose a positive effect of IS development process ambidexterity on system performance in distributed development. Process rigor and process standardization in this research represent two different dimensions of process alignment. Process rigor and process standardization are geared toward improving planned work and "business-as-usual" activities, whereas process agility is geared toward handling unanticipated changes and dynamic events. While rigor and standardization ensure the efficiency and current viability of IS development work, agility ensures the effectiveness and future viability of IS development work. If an IS team focuses on only one of these at the cost of the other, problems will arise and their deliverables will perform poorly. This is particularly important for geographically distributed IS development teams because the communication challenges and higher uncertainty and dynamism imposed by geographic dispersion make both process alignment and agility more important.

Take rigor and agility. While we have proposed that rigor and agility both have main effects on system performance, each may not be as effective in distributed environments in the absence of the other. The benefit of process rigor is relatively limited when process agility is low. Requirement changes in IS development are inevitable. Even if all planned work is done rigorously, the final system may still end up failing because it does not meet changed user requirements. The detrimental consequence of low process agility can overshadow the benefit of process rigor. Conversely, process agility without sufficient process rigor will result in a system that may meet changed user requirements but comes with numerous defects and integration problems. Agility without rigor will cause disorders, confusions, and misunderstandings that will not be easy to repair when communication is hampered due to geographic dispersion. On the other hand, rigorous processes for executing planned work may create a reliable foundation for the software team to effectively respond to requirement changes. This suggests that process rigor and agility may create a synergistic effect and reinforce each other in distributed IS development.

The similar argument can be made for a positive interaction effect of process standardization and agility. In a distributed development environment, many requirement changes would require a concerted effort among multiple locations. If processes are highly standardized across project locations, the coordination cost associated with being agile is likely to be reduced. Conversely, if processes are not standardized across locations and thus idiosyncratic, different processes are used by different locations, being agile will cause high coordination costs.

Our field interview data suggest that change processes become more effective when development processes demonstrate greater rigor and smaller variance across locations resulting from process standardization. When processes are clearly defined, documented, formalized, planned, and standardized, change processes are likely to be well coordinated, reducing the possible negative impact of making changes [30]. Conversely, when processes demonstrate a high level of agility, process rigor and process standardization will help deliver successful systems even when requirements are highly volatile. All these arguments point to a positive interaction effect of process rigor and agility and a positive interaction effect of process standardization and process agility on system performance in distributed development. Therefore, we posit:

H4. Process agility and process rigor have a positive interaction effect on system performance in distributed IS development – i.e., the effect of process agility on system performance is stronger when process rigor is higher and vice versa.

H5. Process agility and process standardization have a positive interaction effect on system performance in distributed IS development – i.e., the effect of process agility on system performance is stronger when process standardization is higher and vice versa.

In this research, we don't hypothesize any significant interaction effect between process rigor and process standardization. They both represent the same alignment dimension of process ambidexterity rather than two different dimensions of it. Therefore, the ambidexterity argument is not applicable for the interaction between the two and we have not found any other theoretical basis for their interaction effect. Their main effects are expected to be additive and thus we expect to find no significant interaction effect between them.

RESEARCH METHODS

Data Collection and Research Sample

Before collecting our primary survey data, we conducted twenty-two field interviews with project managers of distributed IS development projects to formulate research questions, generate measurement items, and help hypothesize relationships between IS process capabilities. We conducted one hour, semi-structured interviews face-to-face and by telephone with project managers from organizations located in countries including Australia, India, the UK, and the USA. These organizations represent the automotive, music, computer, financial, and IT service industries. On average, the interviewees had 6.6 years of project management experience.

We then used a Web-based online survey instrument to collect our primary data. No one from the previous field interviews participated in the online survey. The survey instrument was designed to collect data from two informants from each project – a project manager and a stakeholder (e.g., a project sponsor, a user, or a client) – to avoid common method bias in our data analysis. Project managers responded to items related to process rigor, standardization, and agility of their IS development project as well as the performance of the system delivered by the project. Stakeholders responded to items mainly related to system performance and user satisfaction.

We solicited survey participation from organizations that were partners of an IS research center affiliated with a U.S. private university. We identified and sent an invitation to 171 project managers who had managed IS development projects involving more than one geographic location. In total, we received 103 responses from the project managers. Eleven responses were found to be invalid for this research as they were based on projects involving a single location. Several other responses were found to be incomplete with many missing data or redundant due to multiple data entries. After eliminating such invalid, incomplete, or redundant responses, we retained 85 usable project manager responses for our data analysis, thus resulting in an effective response rate of 49.7%. A large number of our data points came from three large U.S. companies: an oil company, a manufacturing company, and an IT service company. Specifically, 72 projects were from these three companies and 13 projects were from 12 different organizations in various industries.

Then, we contacted project stakeholders of the projects in our sample whose name and contact information were provided by the project managers. We obtained 69 responses from project stakeholders. We found that eight responses were incomplete or redundant and thus eliminated them. As a result, we retained 61 usable responses. Unfortunately, the number of usable stakeholder responses was significantly smaller than that of project manager responses and not deemed large enough to detect hypothesized effects, especially interaction effects. To maximize the statistical power of hypothesis testing, we decided to use only project manager responses for our data analysis. However, stakeholder data were used to examine potential common method bias problems associated with the use of project manager data only.

To assess if significant common method bias existed [53], we performed the following statistical analyses. First, we conducted a Harman's one-factor test [59] on the latent constructs including process rigor, process standardization, process agility, and system performance. Results showed that multiple factors were present and the most covariance explained by one factor is 38.6 percent, indicating that common method biases are not likely to be a serious concern [59]. Second, we tested the consistency between project manager responses and project stakeholder responses on the common items that both responded to. These items included questions about cost overrun, time overrun, and number of system defects. We found fairly high correlations between them, ranging from .44 to .54. Based on the results of the two tests, we concluded that common method biases are unlikely to be a serious problem for this study.

Table 1 shows the characteristics of the research sample. The sample represented different types of IS development projects including new development, off-the-shelf software implementation, and system enhancement. On average, a project team had 55.9 members, a budget of $7.5 million, duration of 16.6 months, indicating that the projects in the sample are relatively large. Structural waterfall development methodologies were used in 44.7% of the projects while 25.0% used agile development methods such as XP and Scrum and 20.3% used hybrid or custom methodologies. All the projects in the sample involved multiple locations: 34.1% were distributed to two to three locations; 30.6% were distributed to four to six locations; and 35.3% were distributed to more than six locations. On average, project managers had about

11 years of project management experience and nearly 20 years of IT-related work experience, indicating that well-rounded and seasoned managers participated in this study.

To examine the possibility of non-response bias, we split the sample into two halves based on the time when each response was received. We compared the early response group with the later response group on variables such as team size, project duration, project type, organizational size, and project management experience. No significant differences between the two subgroups on these variables were found, indicating that non-response bias was not likely to be an issue in this study.

Table 1 about here

Measures

All measurement items used five-point Likert scales to evaluate project manager's perception of the three IS development process capabilities (rigor, standardization, agility) and system performance. The items are shown in Appendix A.

Process Rigor

We measured process rigor using four items. We developed these items based on the CMM literature (e.g., Jalote [42] and Ahern et al. [4]) and based on the results of our field interviews. The items measured the extent to which IS development process was documented and planned in detail, clearly defined and communicated, and formalized.

Process Standardization

We measured process standardization with four items assessing the extent to which common IS development processes were consistently used across multiple sites. Specifically, we measured the consistent use of project management practices, planning methods/techniques, communication methods/technologies, and performance review methods/processes across project sites. These items were identified through our field interviews because no relevant measures for this construct were available from prior research.

Process Agility

We measured process agility with four items assessing how effectively IS development process enabled the team to sense the need for user requirements changes, to strategize its responses, to make decisions to cope with the changes, and to incorporate necessary changes into the system under development. These measurement items were informed by and adapted from prior research (Lee and Xia [49] and Lee and Xia [50]).

System Performance

In this research, system performance is the dependent variable. Informed by prior research [20, 27, 56], we used three items to measure the extent to which the final system delivered by the project had defects, met technical requirements and specifications, and was perceived by the project manager to be a success.

Control Variables

We controlled for geographic dispersion, team size, and project duration as these variables may affect system performance. Geographic dispersion is measured by number of project locations, team size is measured by the number of team members, and project duration is measured in months. Our analysis indicated that all these control variables were distributed in a non-normal fashion. Therefore, we transformed them using a logarithmic transformation to reduce skewness and approximate normality. These transformed data demonstrated normality in Q-Q plots. In addition, we included three binary dummy variables to control for four different groups of data points in the study sample: these four groups include data points from each of the three organizations that accounted for a large portion of the sample and data points from other organizations.

ANALYSIS AND RESULTS

Construct Validity and Reliability

The measurement items of process rigor, process standardization, process agility, and system performance are intended to be reflective measures. Exploratory factor analysis (principal component analysis with Varimax rotation) of these measures produced the intended four-factor structure (see Appendix B). All retained items loaded on their expected factors with loadings ranging from 0.65 to 0.88 and no cross-loadings were found. Table 2 shows means, standard deviations, correlations of the constructs and square root values of the average variance extracted (AVE). Results show that the square root values of AVE for all four constructs are greater than 0.707 and exceed the correlations between the focal construct and other constructs, indicating satisfactory convergent and discriminant construct validities. Cronbach's α values were 0.79 for process rigor, 0.85 for process standardization, 0.83 for process agility and 0.82 for system performance, indicating adequate measurement reliabilities.

Table 2 about here

Regression Model Specification and Estimation Procedure

We chose regression over PLS to test our hypotheses. Prior research suggests that regression is more effective than PLS for testing interaction effects, especially when sample size is less than 200, because the former provides more statistical power than the latter [34]. We specify the following linear equation to test our hypotheses. Since this equation includes interaction terms, we centered all independent variables to minimize multicollinearity between the main effects and interaction terms and to avoid problems associated with lack of scale invariance [5].

$$
\begin{aligned}
\text{System Performance} \\
&= \alpha_0 + \alpha_1 \ln(\text{Geographic Dispersion}) + \alpha_2 \ln(\text{Team Size}) + \alpha_3 \ln(\text{Project Duration}) \\
&+ \alpha_4 \text{Dummy1} + \alpha_5 \text{Dummy2} + \alpha_6 \text{Dummy3} \\
&+ \alpha_7 (\text{Process Rigor}) + \alpha_8 (\text{Process Standardization}) + \alpha_9 (\text{Process Agility}) \\
&+ \alpha_{10} (\text{Process Rigor x Process Agility}) + \alpha_{11} (\text{Process Standardization x Process Agility}) \\
&+ \alpha_{12} (\text{Process Rigor x Process Standardization}) + \varepsilon
\end{aligned}
$$

We estimated the parameters of the equation using a hierarchical OLS (Ordinary Least Squares) regression method. We first entered in the regression model the three control variables and three dummy variables. We then entered the three independent variables and then the three interaction terms subsequently. Although the interaction effect of process rigor and process standardization was not hypothesized in this research, the product term of the two variables was included to make the regression model complete and balanced.

Results of Hierarchical Regression Analysis

The OLS estimates of the unstandardized coefficients of the regression model are presented in Figure 2 and Table 3. The predicted power of the regression model increased significantly as we added to the base model the main effects ($\Delta F = 12.736$, $p < 0.01$) and the interaction effects ($\Delta F = 2.659$, $p = 0.055$). The results of Model 2 provide strong support for H1, H2, and H3, indicating positive main effects of process rigor ($\alpha_7 = 0.296$, $p < 0.01$), process standardization ($\alpha_8 = 0.243$, $p < 0.05$), and process agility ($\alpha_9 = 0.254$, $p < 0.05$) on system performance. The results of Model 3 show consistent support for H1, H2, and H3, although the main effect of process rigor on system performance is slightly weaker ($\alpha_7 = 0.220$, $p = 0.057$). The results of Model 3 also support H4 and show a positive interaction effect of process rigor and process agility on system performance ($\alpha_{10} = 0.301$, $p < 0.05$). However, contrary to our original hypothesis, the results does not support H5 and show a negative interaction effect of process standardization and process agility ($\alpha_{12} = -0.345$, $p < 0.05$) on system performance. Finally, as we expected, no significant interaction effect of process rigor and process standardization is found.

Table 3 about here

Figure 2 about here

Figure 3 illustrates the interaction effects of process rigor, process standardization, and process agility on system performance. We constructed the diagrams in Figure 3 by splitting the data at the median of each variable and calculating the mean of system performance for each of the four subgroups. Therefore, the diagrams are only simplified illustrations of the interaction effects and should not be interpreted as perfectly accurate representations of the interaction effects [28]. When process rigor is low, increased process agility results in relatively small improvement on system performance. When process rigor is high, however, increased process agility results in much larger improvement on system performance (Figure 3(a)). Conversely, when process standardization is low, increased process agility results in relatively large improvement on system performance and when process standardization is high, increased process agility results in much smaller improvement on system performance (Figure 3(b)). The figure illustrates that simultaneous presence of high levels of process agility and process standardization does not lead to highest system performance.

Figure 3 about here

We checked for the presence of multicollinearity using conditions specified by Belsley et al. [10]. The variance inflation factors for the independent variables and the interaction terms in Models 1 to 3 ranged from 1.32 to 2.57. The highest condition index in Models 1 to 3 was 14.79. As significant multicollinearity problems are generally indicated by the variance inflation factor greater than 10 or the condition index greater than 30, the results do not indicate the presence of any serious multicollinearity in our regression results. We conducted Kolmogorov-Smirnov tests to check the normality of the residuals and did not find any violations ($z = 0.75$, $p = 0.62$). We tested for heteroscedasticity of the error terms using White's tests and did not find any violations ($\chi^2_{d.} = 79.4$, $df = 76$, $p = 0.37$).

DISCUSSION

Implications for Theory and Practice

Our research investigated the main and interaction effects of IS development process capabilities on system performance in the geographically distributed development environment. Our results show that all three IS development process capabilities – rigor, standardization, and agility - have a positive main effect on system performance in distributed development. To the best of our knowledge, no prior empirical research has statistically tested such effects in the distributed development environment, though these process capabilities were generally perceived to be valuable for IS development. Thus, our research is one of the first attempts to quantitatively demonstrate the value of rigor, standardization, and agility in distributed IS development.

While prior research tends to suggest a positive effect of process rigor on collocated IS development performance, our research contributes to the literature by extending the generalizability of the effect to the distributed development context. Organizations began to stress the importance of rigorous IS development process long before distributed development became widely adopted. Our research assures that process rigor is critical for successful IS development even when teams are geographically distributed. While some benefits of process standardization in distributed development have been documented based on case studies [58], our research demonstrates such benefits by testing cross-sectional quantitative field data, thus triangulating the finding.

Agile development advocates tend to downplay the importance of process rigor and standardization [9]. However, our results confirm the value of rigor and standardization in distributed IS development. Prior research has documented daunting challenges associated with the effective use of agile processes in distributed IS development [26, 62]. Our results, however, suggest that process agility can actually benefit distributed IS development, which is consistent with recent arguments made in the IS literature [15]. Therefore, we could conclude that process agility may be difficult to build in the context of distributed IS development, but once it is established, it positively affects the performance of distributed development. As IS development is becoming increasingly geographically dispersed, these findings have important implications for advancing theory and practice in IS development. One caution is that we should not confuse process agility with agile methods as the latter does not necessarily guarantee the former and the former can be achieved by other development methodologies [19, 66].

Importantly, our research examined the interaction effects of process alignment (rigor and standardization) and process agility, in addition to their main effects. Understanding the interaction effects of the IS process capabilities is critical because software teams normally seek to build more than one process capability simultaneously. Understanding only their main effects can mislead managers. To the best of our knowledge, no prior research has empirically tested the interaction effects not only in distributed development but also in collocated development. The results showed that process rigor and process agility have a positive interaction effect on system performance above and beyond their respective positive main effects. That is, the ambidexterity of simultaneously increasing process rigor and process agility in distributed development produces a synergistic effect.

Once again, agile software development advocates tend to understate the value of process rigor relative to the value of process agility. Furthermore, process rigor and process agility tend to be viewed as an offsetting relation rather than a synergistic relation. However, our results suggest that the positive effect of process agility on system performance is stronger when the level of process rigor is higher. It appears that rigorous processes for executing planned work create a solid foundation for the team to effectively respond to changes. Conversely, when planned work is not done rigorously, process agility may cause much disorder and chaos. Process rigor may free up the software team's resource and attention for handling dynamic changes. Thus, distributed IS teams are more successful when developing and maintaining both process rigor and agility.

An important question yet to be addressed in further research is "how can a distributed software team simultaneously build both rigor and agility into its development processes?" A challenge is that the practices and methods that foster rigor often weaken agility, and vice versa. For example, agile software development methods value people over processes/tools, working software over comprehensive documentation, and responding to change over following a plan. These agile practices are likely to increase process agility but may decrease process rigor. On the other hand, the implementation of some of the traditional software process improvement methodologies such as CMMI, ISO/IEC 12207, and IEEE/EIA 12207 may increase process rigor at the expense of process agility. Further research needs to investigate these issues to inform researchers and practitioners about effective approaches to building both rigor and agility simultaneously, thus obtaining positive process ambidexterity.

Surprisingly, we found a negative interaction effect of process standardization and process agility on system performance. The results suggest that process standardization and process agility should be characterized as having an offsetting relation rather than a synergistic one, thus requiring an optimal balance between the two. Contrary to the positive interaction of rigor and agility, the ambidexterity of simultaneously increasing standardization and agility is found to reduce system performance in distributed development processes. According to our regression results, on average, when both standardization and agility increase by one unit in scale, the offsetting joint effect produces a decrease of 0.345 in system performance, thus resulting in a total increase of 0.222 in system performance when the main effects (0.347 for agility and 0.220 for standardization) are added. This total increase in system performance is only slightly higher than an increase in system performance solely by one unit increase of process standardization, suggesting that nearly all the main effect of process agility is offset by the negative interaction effect.

One plausible explanation for the negative interaction effect is that process agility may often require local adaptations and improvisations of development processes in certain project locations to effectively respond to idiosyncratic local changes, which makes process standardization across locations incompatible and costly. Conversely, the enforcement of process standardization would make IS development processes tightly-coupled and rigid, which makes local adaptations and changes difficult and costly. Furthermore, process standardization may require high levels of centralization and hierarchical structure which may not be quite compatible with high levels of decentralization and autonomy that process agility might require. These contradictory demands for standardization and agility may cause conflicts and confusion in the IS team's mental model and require the implementation of frequent iterations between adaptation and re-standardization. As a result, the cost associated with the simultaneous pursuit of standardization and agility appears to outweigh the benefit it brings. These offsetting effects are likely to be exacerbated in geographically dispersed environments because distance makes it more difficult to develop common ground and resolve conflicts and issues.

The negative ambidexterity effect of standardization and agility found in our research challenges and modifies the extant organizational ambidexterity literature that proposed only positive ambidexterity effects between alignment and agility. This research suggests that the simultaneous presence of alignment and agility can sometimes lead to no or little increase or even a decrease in organizational performance. Therefore, the effect of organizational ambidexterity on organizational performance appears to be more complex and nuanced than it has been documented by prior literature, at least at the project level, if not at the organizational level.

The fact that two different process capabilities representing process alignment had interaction effects with process agility in opposite direction makes evident the need to analyze these variables at a granular level. It is likely that had we modeled alignment with a single aggregated variable combing rigor and standardization, we would not have found any interaction effects, not because the effects don't exist, but because the interaction effects of the components of alignment would be cancelling each other out. Thus, our study contributes to the literature by illustrating the importance of decomposing alignment variables into more nuanced sub variables. Our speculation is that lateral alignment (e.g., process standardization) and agility are less compatible than vertical alignment (e.g., process rigor) and agility are because agility often causes disturbance to lateral alignment. As a result, the cost of pursuing both lateral alignment and agility is more likely to outweigh its benefit. However, further research is needed to understand and theorize why and under what conditions some types of organizational ambidexterity demonstrate a positive effect and other types of organizational ambidexterity demonstrate a negative effect on organizational performance. One important practical implication of the findings of our research is that, when a project manager evaluates the benefit of a new development practice, he or she needs to assess how it affects not only process rigor but also process standardization and process agility in order to grasp the total effect of the new practice on IS development performance.

Limitations and Conclusions

This research has some limitations. Although we have taken careful steps to validate our measures and to avoid problems of common method variance, all the constructs were measured based on a single informant, the project manager. Although we collected data from project stakeholders as well, the much smaller number of their responses led us to use only the project manager data. Further data analysis is needed to triangulate the results, using multiple data informants. One may argue that even the sample size of project manager data is relatively small. However, the significant main and interaction effects we detected using this sample indicate that these effects are fairly strong. Our study is also limited by the cross-sectional nature of the data. It would be helpful to conduct longitudinal studies with multiple waves of surveys to sort out any spurious or confounding factors. Longitudinal studies can allow us to examine the relative importance of process rigor, standardization, and agility over different phases of the IS development lifecycle and to understand not only the short-term effect of but also the long-term effect of process ambidexterity on IS development performance.

Despite these limitations, our research makes significant contributions to the extant literature and practice. Our research found the positive main effects of three important process capabilities – process rigor, process standardization, and process agility – on system performance in distributed IS development. Further, our research is one of the first attempts to conceptualize and validates the notion of IS development process ambidexterity as an important explanatory factor for system performance in geographically distributed development and demonstrates not only a positive interaction effect of rigor and agility but also a negative interaction effect of standardization and agility on system performance.

Our findings suggest that, contrary to intuition, distributed IS development teams should not blindly apply high levels of rigor, standardization, and agility into their development processes if they wish to maximize system performance. Instead, teams need to be aware of their complex and nuanced main and interaction effects and to strike an informed, delicate balance among them to maximize system performance. All else being equal, for example, our results suggest that increasing both rigor and agility while holding or decreasing standardization produces higher system performance than increasing all three. Further, as the costs associated with building different process capabilities may vary, IS teams need to take these costs into account when finding an optimal balance among them. Future research needs to investigate this cost structure and to provide a more complete picture about the right balance among process rigor, standardization, and agility for distributed IS development.

REFERENCES

1. Adler, P., S.; Goldoftas, B.; and Levine, D.I. Flexibility versus Efficiency? A Case Study of Model Changeovers in the Toyota Production System. *Organization Science*, 10, 1 (1999) 43-68.
2. Agerfalk, P.J., and Fitzgerald, B. Flexible and Distributed Software Processes: Old Petunias in New Bowls? *Communications of the ACM*, 49, 10 (2006) 27-34.
3. Agerfalk, P.J.; Fitzgerald, B.; and Slaughter, S.A. Flexible and Distributed Information Systems Development: State of the Art and Research Challenges. *Information Systems Research*, 20, 3 (2009) 317-328.
4. Ahern, D.M.; Clouse, A.; and Turner, R. *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*. Third Edition ed., Addison-Wesley Professional, 2008.
5. Aiken, L., and West, S. *Multiple Regression: Testing and Interpreting Interactions*. Newbury Park, CA: Sage, 1992.
6. Applegate, L.M.; Austin, R.D.; and McFarlan, F.W. *Corporate Information Strategy and Management*. Boston, MA: Harvard Business School Press, 2007.
7. Athey, S., and Schmutzler, A. Product and Process Flexibility in An Innovative Environment. *Rand Journal of Economics*, 26, 4 (1995) 557-574.
8. Austin, R.D., and Devin, L. Weighing the Benefits and Costs of Flexibility in Making Software: Toward a Contingency Theory of the Determinants of Development Process Design. *Information Systems Research*, 20, 3 (2009) 462-477.
9. Beck, K., and Andres, C. *Extreme Programming Explained: Embrace Change*. 2nd ed., Boston, MA: Addison-Wesley, 2005.
10. Belsley, D.A.; Kuh, E.; and Welsch, R.E. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. New York, NY: Wiley, 1980.
11. Berry, D.; Hungate, C.; and Temple, T. Delivering Expected Value to Users and Stakeholders with User Engineering. *IEEE Software*, 42, 4 (2003) 542-567.
12. Birkinshaw, J., and Gibson, C.B. Building Ambidexterity Into an Organization. *MIT Sloan Management Review*, 45, 4 (2004) 47-55.
13. Boehm, B.W., and Turner, R. *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley, 2004.
14. Carmel, E. *Offshoring Information Technology: Sourcing and Outsourcing to a Global Workforce*. Cambridge, U.K.: Cambridge University Press, 2005.
15. Carmel, E.; Espinosa, J.A.; and Dubinsky, Y. Follow The Sun: Workflow In Global Software Development: Conceptual Foundations. *Journal of Management Information Systems*, 27, 1 (2010) 17-37.
16. Chudoba, K.M.; Wynn, E.; Lu, M.; and Watson-Manheim, M.B. How Virtual Are We? Measuring Virtuality and Understanding its Impact in a Global Organization. *Information Systems Journal*, 15, 4 (2005) 279-306.
17. Cockburn, A. *Agile Software Development*. Boston, MA: Addison-Wesley, 2001.
18. Cockburn, A. *Agile Software Development: The Cooperative Game*. The Second Edition ed., Agile Software Development Series, ed. A. Cockburn and J. Highsmith, Addison-Wesley, 2007.
19. Conboy, K. Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research*, 20, 3 (2009) 329-354.

20. Cooprider, J.G., and Henderson, J.C. Technology-Process Fit: Perspectives on Achieving Prototyping Effectiveness. *Journal of Management Information Systems*, 7, 3 (1991) 67-87.
21. Cramton, C.D. The Mutual Knowledge Problem and its Consequences for Dispersed Collaboration. *Organization Science*, 12, 3 (2001) 346-371.
22. Cummings, J. Work Groups, Structural Diversity, and Knowledge Sharing in a Global Organization. *Management Science*, 50, 3 (2004) 352-364.
23. Cummings, J.; Espinosa, J.A.; and Pickering, C. Crossing Spatial and Temporal Boundaries in Globally Distributed Projects: A Relational Model of Coordination Delay. *Information Systems Research*, 20, 3 (2009) 420-439.
24. Curtis, B.; Kellner, M.; and Over, J. Process Modeling. *Communications of the ACM*, 35, 9 (1992) 75-90.
25. Cusumano, M.A. *Japan's Software Factories: A Challenge to U.S. Management*. New York, NY: Oxford University Press, 1991.
26. Cusumano, M.A. Managing Software Development in Globally Distributed Teams. *Communications of the ACM*, 51, 2 (2008) 15-17.
27. DeLone, W.H., and McLean, E.R. The DeLone and McLean Model of Information Systems Success: A Ten-year Update. *Journal of Management Information Systems*, 19, 4 (2003) 9-30.
28. Espinosa, J.A.; Slaughter, S.A.; Kraut, R.E.; and Herbsleb, J.D. Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development. *Organization Science*, 18, 4 (2007) 613-630.
29. Espinosa, J.A.; Slaughter, S.A.; Kraut, R.E.; and Herbsleb, J.D. Team Knowledge and Coordination in Geographically Distributed Software Development. *Journal of Management Information Systems*, 24, 1 (2007) 135-169.
30. Feldman, M.S., and Pentland, B.T. Reconceptualizing Organizational Routines as a Source of Flexibility and Change. *Administrative Science Quarterly*, 48, 1 (2003) 94-118.
31. Fitzgerald, B. Systems Development Methodologies: The Problem of Tenses. *Information Technology and People*, 13, (2000) 13-22.
32. Fryer, K., and Gothe, M. *Global Software Development and Delivery: Trends and Challenges*, IBM, (2008).
33. Gibson, C.B., and Birkinshaw, J. The Antecedents, Consequences, and Mediating Role of Organizational Ambidexterity. *Academy of Management Journal*, 47, 2 (2004) 209-226.
34. Goodhue, D.; Lewis, W.; and Thompson, R. Statistical Power in Analyzing Interaction Effects: Questioning the Advantage of PLS with Product Indicators. *Information Systems Research*, 18, 2 (2007) 211-227.
35. Gupta, A.K.; Smith, K.G.; and Shalley, C.E. The Interplay between Exploration and Exploitation. *Academy of Management Journal*, 49, 4 (2006) 693-706.
36. Hansen, M., and Lovas, B. How do Multinational Companies Leverage Technological Competencies? Moving from Single to Interdependent Explanations. *Strategic Management Journal*, 25, 801-822 (2004)
37. Harris, M.L.; Collins, R.W.; and Hevner, A.R. Control of Flexible Software Development Under Uncertainty. *Information Systems Research*, 20, 3 (2009) 400-419.
38. Herbsleb, J.D., and Mockus, A. An Empirical Study of Speed and Communication in Globally-Distributed Software Development. *IEEE Transactions on Software Engineering*, 29, 6 (2003) 481-494.
39. Humphrey, W.S. *Managing the Software Process*. Addison-Wesley, 1989.

40. Humphrey, W.S.; Snyder, T.R.; and Willis, R.R. Software Process Improvement at Hughes Aircraft. *IEEE Software*, 8, 4 (1991) 11-23.
41. Im, G., and Rai, A. Knowledge Sharing Ambidexterity in Long-Term Interorganizational Relationships. *Management Science*, 54, 7 (2008) 1281-1296.
42. Jalote, P. *CMM in Practice: Processes for Executing Software Projects at Infosys*. Boston, MA: Addison-Wesley, 2000.
43. Karamouzis, F. *Key Issues for IT Services Sourcing 2011*, Gartner Research Report, (2011).
44. Klimoski, R.J., and Mohammed, S. Team Mental Model: Construct or Metaphor. *Journal of Management*, 20, 2 (1994) 403-437.
45. Kraut, R.; Egido, C.; and Galegher, J. Patterns of Contact and Communication in Scientific Research Collaboration, in J. Galegher, R. Kraut, and C. Egido, ed., *Intellectual Teamwork: Social and Technological Bases of Cooperative Work*, Hillsdale, NJ: Lawrence Erlbaum, 1990, 149-171.
46. Krishnan, M.S., and Kellner, M.I. Measuring Process Consistency: Implications for Reducing Software Defects. *IEEE Transactions on Software Engineering*, 25, 6 (1999) 800-815.
47. Krishnan, V., and Bhattacharya, S. Technology selection and commitment in new product development: The role of uncertainty and design flexibility. *Management Science*, 48, 3 (2002) 313-327.
48. Lee, G.; DeLone, W.H.; and Espinosa, J.A. Flexibility and Rigor: Ambidextrous Coping Strategies in Globally-Distributed Software Development Projects. *Communication of the ACM*, 49, 10 (2006) 35-40.
49. Lee, G., and Xia, W. The Ability of Information Systems Development Project Teams to Respond to Business and Technology Changes: A Study of Flexibility Measures. *European Journal of Information Systems,* 14, 1 (2005) 75-92.
50. Lee, G., and Xia, W. Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility. *MIS Quarterly*, 34, 1 (2010) 87-114.
51. Lewis, M.W. Exploring Paradox: Toward A More Comprehensive Guide. *Academy of Management Review*, 25, 4 (2000) 760-776.
52. Lyytinen, K., and Rose, G.M. Information System Development Agility as Organizational Learning. *European Journal of Information Systems*, 15, 2 (2006) 183-199.
53. Malhotra, N.K.; Kim, S.S.; and Patil, A. Common Method Variance in IS Research: A Comparison of Alternative Approaches and a Reanalysis of Past Research. *Management Science*, 52, 12 (2006) 1865-1883.
54. March, J.G., and Simon, H.A. *Organizations*. New York: Wiley, 1958.
55. Mathieu, J.; Goodwin, G.F.; Heffner, T.S.; Salas, E.; and Cannon-Bowers, J.A. The Influence of Shared Mental Models on Team Process and Performance. *Journal of Applied Psychology*, 85, 2 (2000) 273-283.
56. Nidumolu, S.R. The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable. *Information Systems Research*, 6, 3 (1995) 191-219.
57. Nidumolu, S.R. Standardization, Requirements Uncertainty and Software Project Performance. *Information & Management*, 31, (1996) 135-150.
58. Oshri, I.; van Fenema, P.; and Kotlarsky, J. Knowledge Transfer in Globally Distributed Teams: the Role of Transactive Memory. *Information Systems Journal*, 18, 6 (2008) 593-616.

59. Podsakoff, P.M., and Organ, D.W. Self-Reports in Organizational Research: Problems and Prospects. *Journal of Management*, 12, 4 (1986) 531-544.
60. Pyster, A.B., and Thayer, R.H. Software Engineering Project Management 20 Years Later. *IEEE Software*, September/October (2005) 18-21.
61. Raisch, S., and Birkinshaw, J. Organizational Ambidexterity: Antecedents, Outcomes, and Moderators. *Journal of Management*, 34, 3 (2008) 375-409.
62. Ramesh, B.; Cao, L.; Mohan, K.; and Xu, P. Can Distributed Software Development Be Agile? *Communications of the ACM*, 49, 10 (2006) 41-46.
63. Ramesh, B.; Mohan, K.; and Cao, L. Ambidexterity in Agile Distributed Development: An Empirical Investigation. *Information Systems Research*, (Forthcoming)
64. Royce, W. Successful Software Management Style: Steering and Balance. *IEEE Software*, September/October (2005) 40-47.
65. Sambamurthy, V., and Kirsch, L.J. An Integrative Framework of the Information Systems Development Process. *Decision Sciences*, 31, 2 (2000) 391-411.
66. Sarker, S., and Sarker, S. Exploring Agility in Distributed Information Systems Development Teams: An Interpretive Study in an Offshoring Context. *Information Systems Research*, 20, 3 (2009) 440-461.
67. Standish Group *The Chaos Report*, The Standish Group, (2009).
68. Tiwana, A. Systems Development Ambidexterity: Explaining the Complementary and Substitutive Roles of Formal and Informal Controls. *Journal of Management Information Systems*, 27, 2 (2010) 87-126.
69. Tushman, M.L., and O'Reilly, C.A. Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change. *California Management Review*, 38, 4 (1996) 8-30.
70. Van de Ven, A.H.; Delbecq, L.A.; and Koenig, R.J. Determinants of Coordination Modes within Organizations. *American Sociological Review*, 41, 2 (1976) 322-338.
71. van Oosterhout, M.; Waarts, E.; and van Hillegersberg, J. Change Factors Requiring Agility and Implications for IT. *European Journal of Information Systems*, 15, (2006) 132-145.
72. Vinekar, V.; Slinkman, C.W.; and Nerur, S. Can Agile and Traditional Systems Development Approaches Coexist? An Ambidextrous View. *Information Systems Management*, 23, 3 (2006) 31-42.

Appendix A. Measurement Items for the Project Manager Survey

IS Development Process Capabilities

Please indicate the extent to which you agree or disagree with the following statements
(1: Strongly disagree, 2: Somewhat disagree, 3: Neutral, 4: Somewhat agree, 5: Strongly agree)

Process rigor

1) System requirements were documented in detail (Rigor1)
2) Project team responsibilities were clearly defined and communicated (Rigor2)
3) Project team created a detailed project plan (Rigor3)
4) Project team used a formal software development process (Rigor4)

Process standardization

1) Common project management practices were used consistently across sites (Standard1)
2) Common project planning methods/techniques were used consistently across sites (Standard2)
3) Common communication methods/technologies were used consistently across sites (Standard3)
4) Common project performance review methods/processes were used consistently across sites (Standard4)

Process agility

1) Project team was able to sense user requirements changes effectively (Agility1)
2) Project team was able to strategize its response to user requirements changes effectively (Agility2)
3) Project team was able to make effective decisions to cope with user requirements changes (Agility3)
4) Project team was able to incorporate user requirements changes into the system effectively (Agility4)

System Performance

Please indicate the extent to which you agree or disagree with the following statements
(1: Strongly disagree, 2: Somewhat disagree, 3: Neutral, 4: Somewhat agree, 5: Strongly agree)

1) The system had many defects (SysPerform1)
2) The system met technical requirements/specifications (SysPerform2)
3) The system was a success (SysPerform3)

Appendix B: Factor Loadings Results

Scale item	Process Rigor	Process Standardization	Process Agility	System Performance
Rigor3	**0.76**	0.11	0.10	0.28
Rigor1	**0.73**	0.00	0.25	0.20
Rigor2	**0.73**	0.21	0.18	0.23
Rigor4	**0.72**	0.29	-0.04	0.01
Standard2	0.14	**0.88**	0.16	0.09
Standard1	0.19	**0.84**	0.10	0.10
Standard4	0.27	**0.75**	0.07	0.19
Standard3	0.02	**0.69**	0.25	0.20
Agility3	0.13	0.01	**0.86**	0.03
Agility2	0.11	0.28	**0.81**	0.18
Agility4	0.33	0.07	**0.73**	0.22
Agility1	-0.04	0.31	**0.65**	0.19
SysPerform2	0.12	0.18	0.13	**0.87**
SysPerform3	0.27	0.17	0.16	**0.77**
SysPerform1	0.26	0.15	0.23	**0.73**

Figures

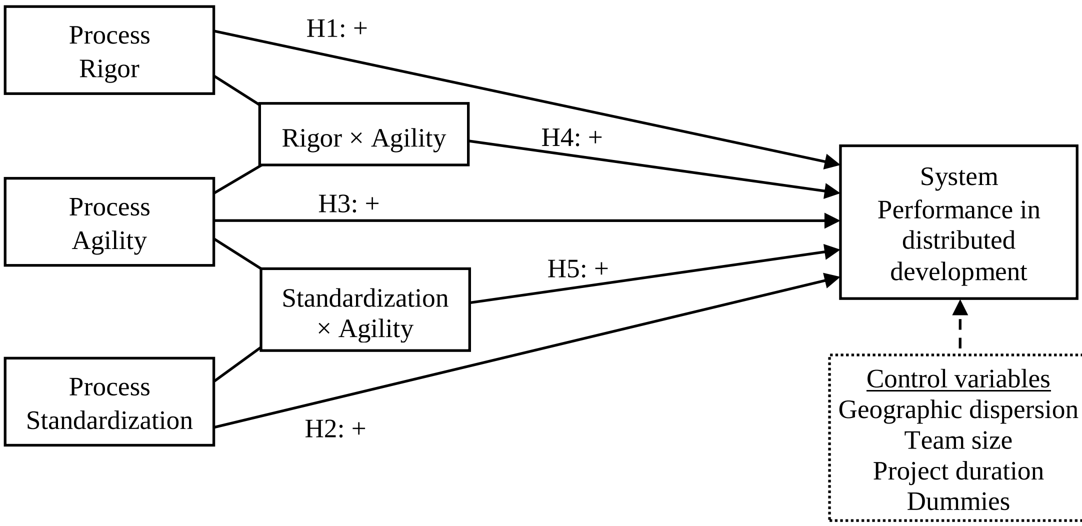
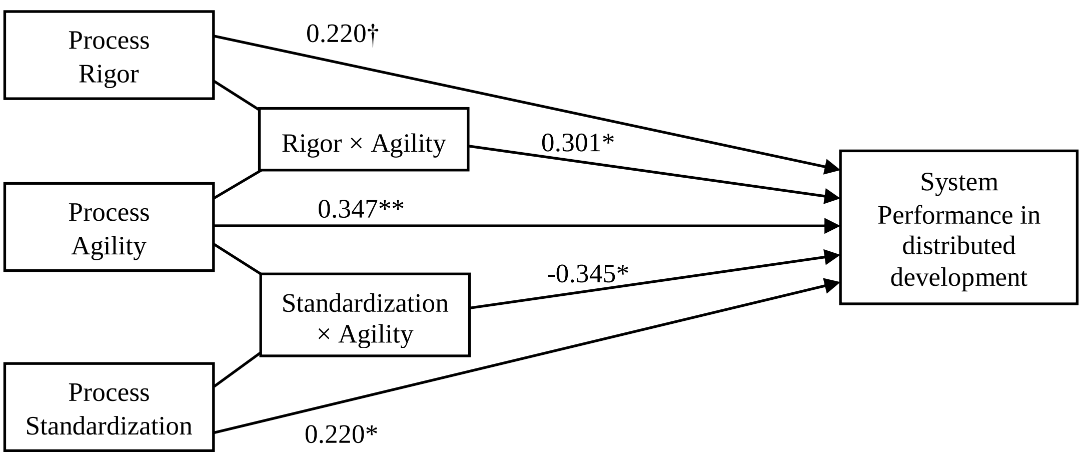
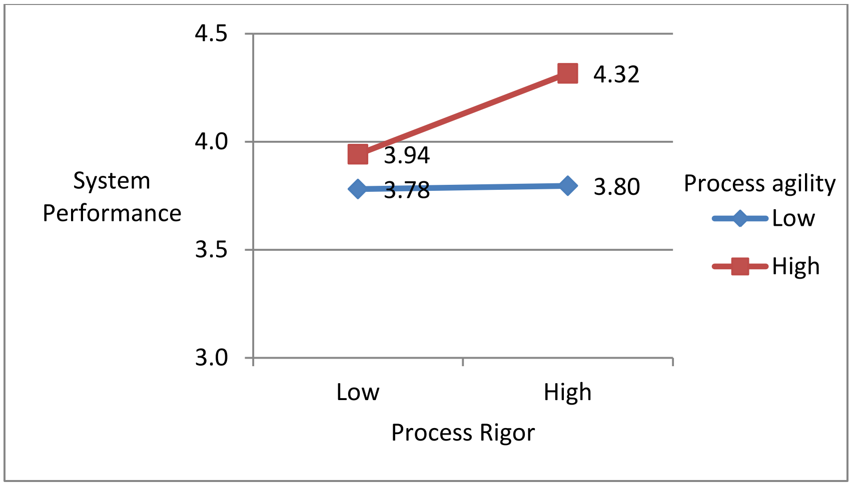


Figure 1 Research Model

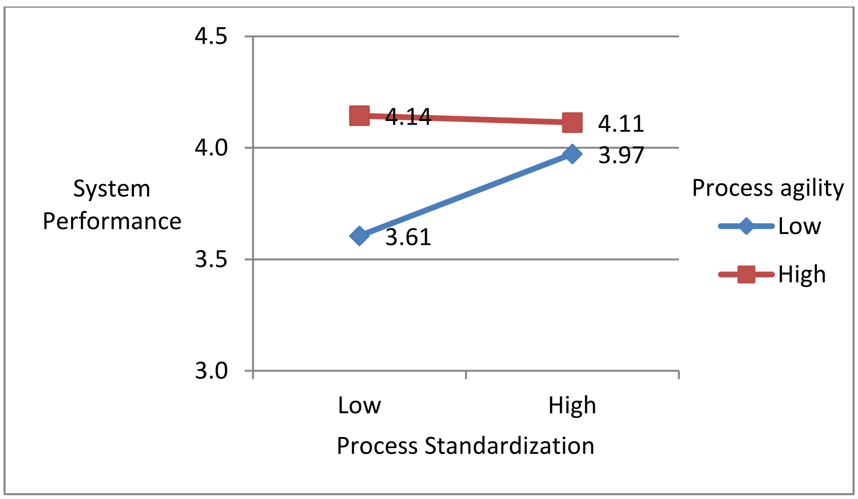


*Note. † p < 0.1; * p < 0.05; ** p < 0.01*

Figure 2 Hierarchical Regression Results(Model 3)



(a) Interaction effect of process rigor and process agility



(b) Interaction effect of process standardization and process agility

Figure 3 Illustration of the Interaction Effects

Tables

Table 1 Sample Characteristics

Variable	Percent	Variable	Percent	Variable	Percent
Project profile		**Project manager profile**		**Organization profile**	
Type of project		Education		Industry	
New development	30.7	College degree	47.1	Manufacturing	14.3
Off-the-shelf software	33.8	Graduate degree	41.2	Software/IT service	32.1
Software enhancement	35.5	Other	11.7	Utility/commodity	40.5
				Bank/finance	2.4
Team size (mean: 55.9)		Gender		Other	8.7
Less than 20 members	37.6	Female	22.4		
20 – 50 members	31.8	Male	77.6	Employees (mean: 116,620)	
Over 50 members	30.6			Less than 10,000	11.5
		Project management experience		10,000 – 100,000	44.2
Budget (mean: $7.5 million)		(mean 11.2)		Over 100,000	44.3
Less than $1 million	32.8	Less than 10 years	40.0		
$1 – 5 million	32.8	10 – 20 years	54.1		
Over $5 million	34.4	Over 20 years	5.9		
Duration (mean: 16.6)		IT work experience (mean 19.9)			
Less than 6 months	11.0	Less than 10 years	10.6		
6-11 months	30.5	10 – 20 years	43.5		
12 – 23 months	31.7	Over 20 years	46.9		
Over 24 months	26.8				
Development methodologies					
Waterfall/structural	44.7				
Agile/iterative	25.0				
Hybrid/custom	20.3				
Number of geographic locations					
2 – 3	34.1				
4 – 6	30.6				
7 – 9	15.3				
10+	20.0				

Notes. n = 85

Table 2 Means, Standard Deviations, and Correlations

Variable	Mean	SD	1	2	3	4	5	6	7
1. *ln* Geographic dispersion	1.59	0.68	–						
2. *ln* Team Size	3.49	1.10	0.35	–					
3. *ln* Project Duration	2.59	0.68	0.28	0.36	–				
4. Process Rigor	3.81	0.70	0.06	-0.11	0.03	0.74			
5. Process Standardization	4.00	0.71	0.08	-0.07	0.06	0.42	0.77		
6. Process Agility	3.78	0.64	0.13	-0.09	0.08	0.37	0.42	0.77	
7. System Performance	3.94	0.71	0.13	-0.10	0.15	0.49	0.43	0.46	0.79

Notes. (1) Correlation coefficients greater than 0.27 are significant at the 0.01 level.

(2) Diagonal elements are the square root of average variance extracted (AVE) by latent constructs from their indicators; Off-diagonal elements are correlations.

Table 3 Results of Hierarchical Regression Analysis on System Performance

Variables	Model 1 Coeffi. (Std. error)	Model 2 Coeffi. (Std. error)	Model 3 Coeffi. (Std. error)
Constant (α_0)	3.932** (0.400)	3.673** (0.383)	3.612** (0.373)
ln Geographic dispersion (α_1)	0.128 (0.141)	0.073 (0.118)	0.028 (0.121)
ln Team Size (α_2)	- 0.107 (0.083)	- 0.049 (0.069)	- 0.029 (0.070)
ln Project Duration (α_3)	0.137 (0.135)	0.151 (0.112)	0.142 (0.110)
Dummy 1 (α_4)	- 0.090 (0.252)	0.091 (0.217)	0.095 (0.210)
Dummy 2 (α_5)	0.176 (0.267)	0.035 (0.231)	0.070 (0.225)
Dummy 3 (α_6)	- 0.234 (0.343)	- 0.227 (0.297)	- 0.127 (0.295)
Process Rigor (α_7)		0.296** (0.107)	0.220† (0.114)
Process Standardization (α_8)		0.243* (0.110)	0.220* (0.108)
Process Agility (α_9)		0.254* (0.114)	0.347** (0.117)
Process Rigor X Process Agility (α_{10})			0.301* (0.151)
Process Standardization X Process Agility (α_{11})			- 0.345* (0.138)
Process Rigor X Process Standardization (α_{12})			- 0.003 (0.123)
F-Statistic	1.527	5.723**	5.242**
Degrees of Freedom	6, 78	9, 75	12, 72
ΔF		12.736**	2.659†
R^2	0.105	0.407	0.466
ΔR^2		0.302	0.059
Adjusted R^2	0.036	0.336	0.377

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