

## Association for Information Systems AIS Electronic Library (AISeL)

ICIS 2010 Proceedings

International Conference on Information Systems  
(ICIS)

2010

# The Main and Interaction Effects of Process Rigor, Process Standardization, and Process Agility on System Performance in Distributed IS Development: An Ambidexterity Perspective

Gwanhoo Lee

*American University*, [glee@american.edu](mailto:glee@american.edu)

William H. DeLone

*American University*, [wdelone@american.edu](mailto:wdelone@american.edu)

J. Alberto Espinosa

*American University*, [alberto@american.edu](mailto:alberto@american.edu)

Follow this and additional works at: [http://aisel.aisnet.org/icis2010\\_submissions](http://aisel.aisnet.org/icis2010_submissions)

### Recommended Citation

Lee, Gwanhoo; DeLone, William H.; and Espinosa, J. Alberto, "The Main and Interaction Effects of Process Rigor, Process Standardization, and Process Agility on System Performance in Distributed IS Development: An Ambidexterity Perspective" (2010). *ICIS 2010 Proceedings*. 34.

[http://aisel.aisnet.org/icis2010\\_submissions/34](http://aisel.aisnet.org/icis2010_submissions/34)

This material is brought to you by the International Conference on Information Systems (ICIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICIS 2010 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# **The Main and Interaction Effects of Process Rigor, Process Standardization, and Process Agility on System Performance in Distributed IS Development: An Ambidexterity Perspective**

*Completed Research Paper*

**Gwanhoo Lee**

Kogod School of Business  
American University  
4400 Massachusetts Ave., NW  
Washington, D.C. 20016  
glee@american.edu

**William H. DeLone**

Kogod School of Business  
American University  
4400 Massachusetts Ave., NW  
Washington, D.C. 20016  
wdelone@american.edu

**J. Alberto Espinosa**

Kogod School of Business  
American University  
4400 Massachusetts Ave., NW  
Washington, D.C. 20016  
alberto@american.edu

## **Abstract**

*Information systems (IS) development is becoming increasingly more geographically dispersed. Although process rigor, process standardization, and process agility are generally believed to have a positive impact on software development, it has not been well understood how these process capabilities affect distributed IS development. More important, no prior research has investigated their interaction effects. Drawing upon prior literature on organizational ambidexterity, we hypothesize: positive main effects of process rigor, process standardization, and process agility; a positive interaction effect of process rigor and process agility; and a positive interaction effect of process standardization and process agility on system performance in distributed development. Our data analysis results support a positive main effect of the three process capabilities. We find a positive interaction effect of process rigor and process agility suggesting positive process ambidexterity of rigor and agility. Surprisingly, we find a negative interaction effect of process agility and process standardization suggesting negative process ambidexterity of agility and standardization.*

**Keywords:** Process rigor; process agility; process standardization; distributed IS development; system performance; interaction effects; process ambidexterity

## Introduction

Information systems (IS) development is becoming increasingly more geographically dispersed in part due to globalization, outsourcing, and offshoring trends (Espinosa, Slaughter, Kraut, and Herbsleb 2007). While taking advantage of lower costs and broader talent pools, organizations face new challenges associated with distributed development (Cusumano 2008). Geographic dispersion causes higher complexity and dynamism of software development which in turn make it difficult for software teams to coordinate and carry out their work (Herbsleb and Mockus 2003; Xia and Lee 2005). Therefore, the growing adoption of distributed development may exacerbate historically-low success rate of IS development (Standish Group 2009).

Prior research has consistently supported the importance of process capability in software development (Sambamurthy and Kirsch 2000). Over the last several decades, organizations emphasized a few different types of process capabilities as critical success factors for software development. In the 1980s and the 1990s, organizations have focused their efforts mostly on improving process rigor (Pyster and Thayer 2005). Numerous system development methodologies and approaches such as the Waterfall Model and the Capability Maturity Model were used to improve the rigor of software process both at the project level and at the organizational level (Harter, Krishnan, and Slaughter 2000; Jalote 2000; Krishnan, Kriebal, Kekre, and Mukhopadhyay 2000). These structured, plan-driven development approaches were widely implemented to reduce defects, improve quality and productivity, and increase user satisfaction. However, they often led to project failures as they did not allow software teams to change freely from the detailed plan laid out prior to the project's start (Austin and Devin 2003).

Process standardization has long been recommended as the way to improve software development performance (Nidumolu 1996). In distributed development, organizations may benefit from process standardization across sites because standardization helps overcome differences derived from local contexts (Oshri, van Fenema, and Kotlarsky 2008). Distributed development decreases communications and increases the possibilities for conflict, misunderstanding, and breakdowns in communication (Chudoba, Wynn, Lu, and Watson-Manheim 2005). It is very challenging for distributed teams to develop shared understanding because members of such teams do not stand on 'common ground' (Cramton 2001). Standardized processes can facilitate communication and coordination among team members distributed geographically who, otherwise, might have very different ways of carrying out tasks (Oshri et al. 2008). Furthermore, standardized processes can reduce the negative impact of personnel turnover and lead to a cohesive organizational culture (Humphrey, Snyder, and Willis 1991; Nidumolu 1996).

In the late 1990s and the early 21st century, recognizing the detrimental consequences of the lack of process adaptability in software development, organizations began to adopt agile development approaches such as XP (Extreme Programming) and Scrum to improve software development flexibility in responding to ever-changing requirements (Beck and Andres 2005; Cockburn 2007). The agile camp emphasizes process agility over process rigor and standardization and implies that overly rigorous and standardized processes do not pay off. Although agile processes generally benefit small-scale and collocated development, it is questionable if such processes are applicable and useful for distributed development (Cusumano 2008).

Some might argue that process rigor and process agility are conflicting and incompatible and thus undermine development performance when they are present simultaneously. However, others propose that IS development often require both rigor and agility instead of requiring one over the other (Applegate, Austin, and McFarlan 2007; Boehm and Turner 2004; Pyster and Thayer 2005). The same argument can be true for process standardization and process agility. Therefore, distributed IS development appears to face paradoxical requirements for software process (Ramesh, Cao, Mohan, and Xu 2006). On the one hand, it requires rigor, formality, and standardization in its process to overcome communication and coordination difficulties resulting from team boundaries of distance, time, culture, and organization (Lee, DeLone, and Espinosa 2006). On the other hand, it requires agility to cope with high uncertainty and dynamism exacerbated by geographic dispersion (Royce 2005).

This research aims to fill the gap in prior literature by addressing the following issues. First, although prior literature argues that process rigor, process standardization, and process agility generally have positive effects on IS development, no research has empirically tested their effects in the context of distributed development. In this research, we empirically test if all three process capabilities indeed have positive effects on system performance in distributed development. By doing so, we validate the usefulness of the three process capabilities in the context of distributed development. Second, we examine if and how the three process capabilities interact with one another to affect system performance in distributed development above and beyond their main effects. Prior research has not

addressed this critical question. Drawing upon the organizational ambidexterity literature, we hypothesize these interaction effects and empirically test them with cross-sectional survey data. This research significantly advances the extant literature by uncovering the interaction effects of process capabilities on distributed development. Furthermore, this research can provide IS project managers with a framework for evaluating the effectiveness of various system development methodologies in terms of rigor, standardization, and agility.

The remainder of the paper is organized as follows. In the next section, we discuss theoretical background for this research and develop our hypotheses based on prior literature on IS development process, distributed software teams, organizational ambidexterity, and IS project performance as well as based on field interview data from IS projects managers. We then discuss our research methods and test the hypotheses using our survey data. Finally, we present the results and conclude this paper with the discussion on the implications of the findings for theory and practice.

## Hypotheses Development

The effectiveness of IS development processes has been proposed to be critical for IS project performance (Sambamurthy and Kirsch 2000). An IS development process is defined as a set of activities performed during the development of a system (Curtis, Kellner, and Over 1992). These activities involve interactions among people, technology, methods, and procedures (Humphrey 1989). Over the last several decades, the software development literature has identified a few important process capabilities including process rigor, process standardization, and process agility. These process capabilities deem to be effective in general. However, no research has examined their effects on IS development performance in distributed environment.

The recent development of the organizational ambidexterity perspective can be used as a useful theoretical lens to theorize the interaction effects of different software process capabilities. Organizational ambidexterity is defined as an organization's ability to concurrently demonstrate alignment and adaptability (Gibson and Birkinshaw 2004), exploitation and exploration (Gupta, Smith, and Shalley 2006; Im and Rai 2008), efficiency and flexibility (Adler, Goldoftas, and Levine 1999) or incremental change and revolutionary change (Tushman and O'Reilly 1996). The dichotomy between alignment and adaptability appears to be particularly relevant to IS development process capabilities. While some IS development process capabilities tap into alignment aspects, other process capabilities tap into adaptability aspects. The alignment dimension of organizational ambidexterity refers to the coherence among all the patterns of activities in the business unit, and adaptability refers to the organizational capacity to tailor activities in the business unit quickly to meet changing demands in the task environment (Gibson and Birkinshaw 2004).

Drawing upon the literature on software development, distributed software teams, and organization ambidexterity as well as our field interview data, we develop the following five hypotheses with respect to the main effects and the interaction effects of process rigor, process standardization, and process agility on system performance in distributed development. Figure 1 shows our research model.

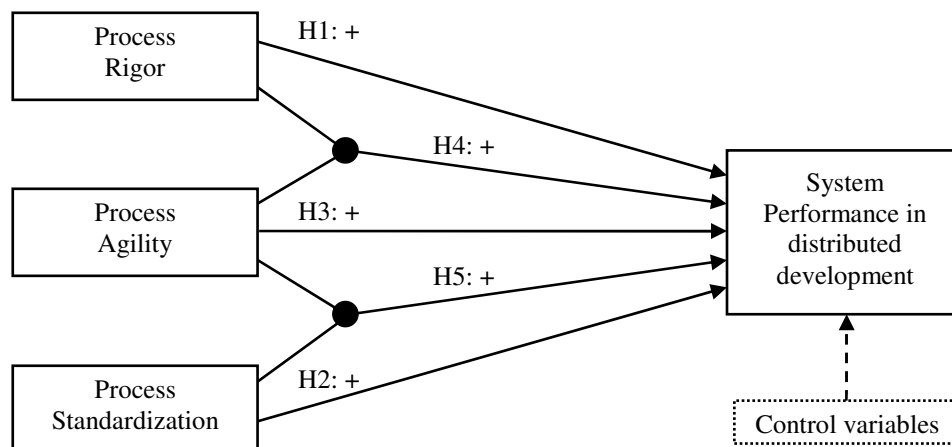


Figure 1. Research Model

## ***Main Effects of Process Rigor, Process Standardization, and Process Agility on System Performance***

We define process rigor as the strict adherence, compliance, and precision to pre-defined, formal, and structured IS development process. As such, process rigor is introduced and built through clear, formal, and detailed definitions of roles, activities, work products, methods, and measures related to software development (Applegate et al. 2007; Berry, Hungate, and Temple 2003). Process rigor is characterized by detailed system requirements documentation, thorough review of system requirements by key stakeholders, clear definition of responsibilities and goals, detailed project plans, formal estimation methods, formal development phases, accurate performance tracking, and frequent progress review and measurement (Jalote 2000; Lee et al. 2006).

Process rigor is especially emphasized by plan-driven, structured approaches to IS development (Boehm and Turner 2004) and is exemplified by the Capability Maturity Model Integrated (CMMI) (Ahern, Clouse, and Turner 2008; Krishnan and Kellner 1999; Pyster and Thayer 2005). Process rigor is not easily achieved in practice as one study shows that only 6% of software developers rigorously adhere to methods (Fitzgerald 2000). Process rigor helps distributed development overcome communication impedance (Ramesh et al. 2006). The lack of process rigor is often responsible for IS development failures (Pyster and Thayer 2005). We argue that process rigor helps distributed software teams overcome communication impedance and effectively coordinate development tasks because well-defined processes reduce task-related ambiguity. 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 that make projects successful from my standpoint.”* Without rigorous IS development processes, distributed teams are more likely to make errors and mistakes, resulting in integration problems and lower system performance. Therefore, we posit:

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

We define process standardization as the uniformity, commonality, and consistency of IS development processes across development sites. We argue that standardized processes enable team members to understand each other’s work easily, coordinate their activities efficiently, and undertake system integration effectively. Standardized processes save a great deal of time and effort especially in the inception of a distributed development project because distributed teams otherwise would need to spend a significant amount of time and effort to set up ground rules and protocols for task coordination.

Consistent processes enable team members to better understand each others’ work, coordinate their activities more easily, and thus undertake system integration more effectively (Lee et al. 2006; Nidumolu 1996). 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 local contexts (Oshri et al. 2008). Process standardization helps distributed software teams to develop a shared mental model of how work is done (Klimoski and Mohammed 1994; Mathieu, Goodwin, Heffner, Salas, and Cannon-Bowers 2000). Standardized processes lead to a cohesive organizational culture through the common technical language, procedures, and goals (Humphrey et al. 1991).

One project manager in our field interview stated, *“One of the key things we have done right is, we have internally developed project management automation 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 posit that standardized IS development processes across different locations lead to system performance by reducing process variance. 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 development.*

Consistent with prior literature (Lee and Xia 2005; Lee and Xia 2010; Lyytinen and Rose 2006; Tiwana, Bharadwaj, and Sambamurthy 2003), we define process agility as the process capability to effectively sense and respond to changing system requirements. The rapid adoption of agile development methods suggests that process agility is perceived by organizations to be an important capability for IS development. Process agility enables software teams to strategize their responses to changing system requirements and implement necessary changes in a timely and cost-

effective manner (Beck and Andres 2005; Cockburn 2001). Process agility lowers the cost of incorporating changes (Athey and Schmutzler 1995). Due to inherent uncertainty in business and technology environments, changes in system requirements are inevitable (Krishnan and Bhattacharya 2002). Distributed development environment exacerbates the uncertainty of system requirements because of difficulties in defining initial requirements. Furthermore, requirements are more volatile due to competitive environment changes occurring across multiple team boundaries. Thus, handling these requirement changes imposes a difficult challenge on distributed software teams.

System performance would be significantly undermined when IS development processes are not designed to effectively manage requirement changes. 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."* Process agility is needed especially when required changes were not anticipated or adequately specified at the onset of the project (van Oosterhout, Waarts, and van Hillegersberg 2006). We posit that a streamlined, efficient process that enables the software team to sense critical requirement changes and to strategize and implement appropriate responses is critical to system performance in geographically distributed development.

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

### ***Interaction Effects of Process Rigor, Process Agility, and Process Standardization on System Performance***

The organizational ambidexterity literature conceptualizes ambidexterity to be the simultaneous presence of alignment and adaptability. Similarly, we define IS development process ambidexterity as the process capability that simultaneously demonstrates alignment and adaptability in IS development. In this research, we conceptualize process rigor and process standardization as two different types of process alignment. While process rigor focuses on clarity, formality, and exactness of IS development processes, process standardization mainly focuses on uniformity and consistency of IS development processes across development sites. Therefore, they capture different dimensions of process alignment in the distributed development context. On the other hand, we conceptualize process agility as process adaptability in this research. Process agility refers to the process capability to efficiently and effectively respond to changing user requirements (Lee and Xia 2010). Therefore, agility requires adaptation of and departure from planned actions and processes.

Derived from the organizational ambidexterity literature, we propose that the simultaneous presence of process alignment and process adaptability positively affects system performance in distributed development. Although several researchers maintain that there is an inherent trade-off between organizational alignment and adaptability (Ancona, Goodman, Lawrence, and Tushman 2001; Floyd and Lane 2000; Levinthal and March 1993), the possibility of achieving both alignment and adaptability has been suggested by recent literature (Gibson and Birkinshaw 2004; Im and Rai 2008; Tushman and O'Reilly 1996). Process rigor and process standardization are geared toward improving routines and "business-as-usual" activities. Process agility, on the other hand, is geared toward handling unanticipated changes and dynamic events. If a software team focuses one of these at the cost of the other, problems will arise. We argue that the most successful software teams reconcile the tension between process rigor/process standardization and process agility and, by doing so, enhance their software development performance. Our field interview data suggest that change processes become more effective when development processes demonstrate greater rigor and smaller variance. When processes are well-defined, well-documented, formalized, detailed, and standardized, change processes can be well coordinated, reducing the possible negative impact of changes on system performance. Therefore, we posit:

*H4. The higher the levels of process rigor and process agility, the higher the level of system performance.*

*H5. The higher the levels of process standardization and process agility, the higher the level of system performance.*

## Method

### *Data Collection and Research Sample*

Before collecting survey data, we conducted twenty-two field interviews with project managers of global IS projects to help formulate research questions, generate measurement items, and theorize relationships among process capabilities. We conducted one hour, semi-structured interviews face-to-face and by telephone with several managers from organizations located in various 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 field interviews participated in the survey. The survey instrument was designed to collect data from two participants from each project, a project manager and a stakeholder (e.g., a project sponsor, user, or client of the project), to avoid common method bias. As shown in Appendix, project managers responded to questions related to process rigor, process standardization, and process agility of their software development and system performance. On the other hand, stakeholders only responded to questions related to system performance.

We solicited survey participation from organizations that were partners of an IS research center affiliated with a U.S.-based private university. We identified and sent an invitation to 171 project managers who had experience managing IS development projects involving more than one geographic location. In total, we received 103 project manager responses. After eliminating invalid, incomplete, or redundant responses, we retained 85 usable project managers' 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 the number of project manager responses and not deemed large enough to detect effects. To maximize the statistical power of hypothesis testing, we decided to use only project manager responses for our data analysis. Stakeholder data were used to examine potential common method bias problems of project manager data as well as to provide additional insights beyond our main results.

To assess if potential common method bias was a significant issue (Malhotra, Kim, and Patil 2006), we performed the following statistical analyses. First, we conducted a Harman's one-factor test (Podsakoff and Organ 1986) on the latent constructs including process rigor, process standardization, process agility, and system performance. Results showed that multiple factors are 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 (Podsakoff and Organ 1986). Second, we tested the consistency between responses between project managers and project stakeholders on the common questionnaire items that both responded to. These items included questions about time cost overrun, time overrun, and number of system defects. We found relatively high correlations between them, ranging from .44 to .54.

Table 1 shows the characteristics of the research sample. The sample included different types of information systems development projects including new development (26.3%), off-the-shelf software implementation (28.9%), system enhancement (30.4%), and others (14.4%). On average, a project team had 55.9 members, a budget of \$7.5 million, duration of 16.6 months. 44.7% of projects used structural waterfall development methodologies while 25.0% used agile development methods. All the projects in the sample involved multiple locations. Project managers had about 11 years of project management experience and nearly 20 years of IT-related work experience, indicating that the survey includes well-rounded and seasoned managers. To examine the possibility of non-response bias, we split the sample into two half-sized sub-groups based on the time when each response was received. We then 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 groups on these variables were found, indicating that non-response bias was not likely to be an issue.

**Table 1. Sample Characteristics**

Variable	Percent	Variable	Percent	Variable	Percent
<b>Project profile</b>		<b>Project manager profile</b>		<b>Organization profile</b>	
Team size (mean: 55.9)		Project management experience (mean 11.2)		Industry	
Less than 20 members	37.6	Less than 10 years	40.0	Manufacturing	14.3
20 – 50 members	31.8	10 – 20 years	54.1	Software/IT service	32.1
Over 50 members	30.6	Over 20 years	5.9	Utility/commodity	40.5
Budget (mean: \$7.5 million)		IT work experience (mean 19.9)		Bank/finance	2.4
Less than \$1 million	32.8	Less than 10 years	10.6	Other	8.7
\$1 – 5 million	32.8	10 – 20 years	43.5	Employees (mean: 116,620)	
Over \$5 million	34.4	Over 20 years	46.9	Less than 10,000	11.5
Duration (mean: 16.6)				10,000 – 100,000	44.2
Less than 6 months	11.0			Over 100,000	44.3
6-11 months	30.5				
12 – 23 months	31.7				
Over 24 months	26.8				

Notes. n = 85

### **Measures**

All measurement items for the three IS development process capabilities and system performance used five-point Likert scales. These measures were designed to evaluate project manager's perception of project process and performance. The items are shown in Appendix.

### **Process rigor**

We measured process rigor using four items. We developed these items based on the CMM literature (e.g., Jalote (2000) and Ahern et al. (Ahern et al. 2008)) and based on the results of our field interviews. The items intend to measure the extent to which IS development process is defined, communicated, documented, formalized, planned, and detailed.

### **Process standardization**

The items for process standardization intend to measure the extent to which common IS development processes are consistently used across project sites. With four items, we measured the consistent use of project management practices, planning methods/techniques, communication methods/technologies, and performance review methods/processes. These items were selected based on comments from our field interviews.

### **Process agility**

Using four items, we measured process agility by assessing how effectively IS development process enables the team to sense the need for requirements changes, plan its responses, and incorporate necessary changes into the system under development. These measurement items were informed by and adapted from Lee and Xia (2005).

### **System performance**

In this research, system performance is the only dependent variable. With three items, we measured the extent to which the final system delivered by the project had defects, met technical requirements and specifications, and were perceived by project managers to be a success.



## Control and dummy variables

We controlled for variables that may affect system performance, such as team size and project duration. Team size is measured by the number of team members and project duration is measured in months. Our analysis indicated that these three control variables were distributed in a non-normal fashion. Therefore, we transformed the variables using a logarithmic transformation to reduce skewness and approximate normality. In addition, we controlled for dummy variables for three organizations in the study sample that accounted for a large portion of data points.

## Analysis and Results

### Construct Validity and Reliability

The measurement items of process rigor, process standardization, process agility, and system performance in this research are designed to be reflective measures. Exploratory factor analysis (principal component analysis with Varimax rotation) of these measures produced the intended four-factor structure. All retained items loaded on their expected factors with loadings ranging from 0.65 to 0.88. Table 2 shows correlations of the constructs and square root values of the average variance extracted (AVE) and. Results show that the AVE values for all four constructs are greater than 0.707 and exceed the correlations with other constructs, indicating satisfactory convergent and discriminant validities for the constructs. Cronbach's  $\alpha$  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 construct reliabilities.

**Table 2. Correlations and Average Variance Extracted**

Variable	1	2	3	4	5	6
1. <i>ln</i> Team Size	–					
2. <i>ln</i> Project Duration	0.36**	–				
3. Process Rigor	-0.11	0.03	0.74			
4. Process Standardization	-0.07	0.06	0.42**	0.77		
5. Process Agility	-0.09	0.08	0.37**	0.42**	0.77	
6. System Performance	-0.10	0.15	0.49**	0.43**	0.46**	0.79

Notes. (1) \*  $p < 0.05$ ; \*\*  $p < 0.01$ .

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

### Model Specification and Estimation Procedure

We specify the following linear equation to test our hypotheses. Since this equation includes interaction terms, we centered all independent variables to avoid possible multicollinearity and scale invariance problems and to provide a more accurate interpretation of effects (Aiken and West 1992).

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

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

### Results of Estimation

The OLS estimates of the parameters of the equation are provided in Table 3. The predicted power of the regression model increased significantly as we added to the base model the main effects ( $\Delta F = 13.116$ ,  $p < 0.01$ ) and interaction effects ( $\Delta F = 2.813$ ,  $p < 0.05$ ) of our hypotheses. The estimates of the parameters shown in Model 3 provide support for H1 to H4. However, H5 was not supported, demonstrating a negative effect instead of a hypothesized positive effect. More specifically, the positive main effects of process rigor ( $\alpha_6 = 0.225$ ,  $p < 0.05$ ), process standardization ( $\alpha_7 = 0.217$ ,  $p < 0.05$ ), and process agility ( $\alpha_8 = 0.348$ ,  $p < 0.01$ ) on system performance are statistically significant. The results support the positive interaction effect of process rigor and process agility ( $\alpha_9 = 0.298$ ,  $p < 0.05$ ). However, the results show a negative interaction effect of process standardization and process agility ( $\alpha_{12} = -0.351$ ,  $p < 0.05$ ) on system performance.

We checked for the effect of multicollinearity using conditions specified by Belsley et al. (Belsley, Kuh, and Welsch 1980). 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 13.25. Thus, the results do not indicate the presence of any serious multicollinearity effects in our regression models. We conducted Kolmogorov-Smirnov tests to check the normality of the residuals and did not find any violations ( $z = 0.68$ ,  $p = 0.75$ ). We tested for heteroscedasticity of the error terms using White's tests and did not find any violations ( $\chi^2_{d.f.} = 76.6$ ,  $df = 77$ ,  $p = 0.49$ ).

**Table 3. Results of Hierarchical Regression Analysis on System Performance**

Variables	Coefficients (unstandardized)		
	Model 1	Model 2	Model 3
Constant ( $\alpha_0$ )	3.738**	3.560**	3.571**
ln Team Size ( $\alpha_1$ )	-0.080	-0.033	-0.023
ln Project Duration ( $\alpha_2$ )	0.142	0.153	0.143
Dummy 1 ( $\alpha_3$ )	-0.016	0.136	0.110
Dummy 2 ( $\alpha_4$ )	0.293	0.103	0.094
Dummy 3 ( $\alpha_5$ )	-0.117	-0.155	-0.100
Process Rigor ( $\alpha_6$ )		0.304**	0.225*
Process Standardization ( $\alpha_7$ )		0.236*	0.217*
Process Agility ( $\alpha_8$ )		0.258*	0.348**
Process Rigor X Process Agility ( $\alpha_9$ )			0.298*
Process Standardization X Process Agility ( $\alpha_{10}$ )			-0.351*
Process Rigor X Process Standardization ( $\alpha_{11}$ )			-0.004
F-Statistic	1.671	6.443***	5.789***
$\Delta F$		13.116***	2.813*
$R^2$	0.096	0.404	0.466
$\Delta R^2$		0.309	0.062

Notes. \*  $p < 0.05$ ; \*\*  $p < 0.01$ ; \*\*\*  $p < 0.001$ .

## Discussion

### *Implications for Theory and Practice*

The regression results generally supported our hypotheses. However, contrary to our expectation, we found a negative interaction effect of process standardization and process agility on system performance in distributed development. We discuss the implications of these results, followed by limitations and conclusions.

All three IS development process capabilities – rigor, standardization, and agility - demonstrated a positive effect on system performance. Although the positive effects of these process capabilities on IS development have been proposed by prior literature, no research has provided empirical evidence of these effects in the context of distributed IS development. Thus, our research is one of the first attempts to confirm the value of process rigor, process standardization, and process agility in distributed IS development. Our research informs IS project managers of the importance of each of the three process capabilities in distributed development. Developing these process capabilities generally increases the likelihood of system success in distributed development projects.

More important, our research examines the interaction effects of the process capabilities. To the best of our knowledge, no prior research has empirically tested them. In particular, based on the organizational ambidexterity perspective, we posited a positive interaction effect of process rigor and process agility and a positive interaction effect of process standardization and process agility on system performance in distributed development. The results supported that process rigor and process agility indeed have a positive interaction effect above and beyond their respective main effects on system performance. That is, the positive effect of process ambidexterity of rigor and agility was validated for the context of distributed IS development. Agile development advocates tended to understate the value of process rigor relative to the value of process agility. Furthermore, process rigor and process agility tended to be viewed as a trade-off. However, our research suggests that process rigor makes process agility even more effective and that distributed software teams are better off by developing and maintaining both process capabilities. Our research shows that, at least in distributed development, process rigor and process agility demonstrate a synergistic effect on development performance. This finding significantly advances the body of knowledge on distributed software development and warrants further research in this area. One of the practical implications of this research is that when a project manager evaluates the effectiveness of a new system development methodology, he/she needs to understand how the new methodology affects not only process rigor but also process standardization and process agility.

Then, an important question to be addressed in future research is “How can both process rigor and process agility be simultaneously achieved?” The practices and methods that foster process rigor often undermine process 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 would increase process rigor but may decrease process agility. Future research needs to address these issues to inform researchers and practitioners about effective approaches to building both process rigor and process agility, thus demonstrating process ambidexterity.

Surprisingly, we found a negative interaction effect of process standardization and process agility on system performance whereas we originally hypothesized a positive interaction effect. The results suggest that the relationship between process standardization and process agility should be characterized by a trade-off. Although our data cannot provide a definite explanation for this unexpected finding, our best speculation is that process agility may require local adaptations of development processes in certain locations to effectively respond to idiosyncratic changes, which contradicts the requirement for global process standardization. In other words, process standardization tends to enforce tightly-coupled processes where local changes are costly. These contradictory demands may cause confusion in the mental model, process reconfiguration, and thus inefficiency in IS development.

Although both process rigor and process standardization are conceptualized as process alignment from the ambidexterity perspective, their interaction effects with process agility are found to have opposite directions. In fact, the negative ambidexterity effect of standardization and agility found in this research may have significant implications for theory and practice. Contrary to the extant organizational ambidexterity literature that proposed

only the positive effect of ambidexterity, this research suggests that the simultaneous presence of alignment and adaptability can sometimes lead to inferior organizational performance. We argue that the effect of ambidexterity on performance may be more complex and nuanced than it has been documented by prior literature. Practitioners need to be aware of this trade-off relationship between process standardization and process agility and seek the right balance that maximizes development performance. Further research is warranted to better understand and theorize why some types of process ambidexterity demonstrate a positive effect and other types of process ambidexterity demonstrate a negative effect on distributed IS development.

### ***Limitations and Future Research***

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 forced us to use data solely from project managers. Further data analysis is needed to triangulate the results, using multiple sources of informants. Our study is also limited by the cross-sectional nature of our data. It would be useful to conduct longitudinal studies with multiple waves of surveys to reveal the process by which process ambidexterity is achieved. Longitudinal studies can allow us to examine the relative importance of process rigor, process standardization, and process agility over different phases of the 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 study is one of the first attempts to conceptualize and validates the notion of IS development process ambidexterity as an important explanatory factor for the performance in geographically distributed development. Furthermore, our study demonstrates not only the positive effect of but also the negative effect of process ambidexterity on system performance in distributed development. As a result, this research further advances theories of distributed IS development and of organizational ambidexterity and provides useful insights to practitioners about the right balance among process rigor, process standardization, and process agility.

Working on highly interdependent tasks across very complex team boundary configurations with rapidly changing requirements is a daunting task for software teams. Our study shows that ambidexterity of process rigor and process agility is one possible answer to help distributed software teams achieve higher levels of system performance. While our results are useful and stimulating, further research in this area is needed to deeply understand the complex effect of process ambidexterity.

## **Appendix 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)

### **Acknowledgements**

We thank the financial support from Center for IT and the Global Economy (CITGE) in the Kogod School of Business at American University, Washington, DC. We thank Professors Jungpil Hahn, Ghiyoung Im, and Arun Rai for their helpful comments on the earlier versions of this paper.

### **References**

- 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, pp 43-68.
- Ahern, D.M., Clouse, A., and Turner, R. *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*, (Third Edition ed.) Addison-Wesley Professional, 2008.
- Aiken, L., and West, S. *Multiple Regression: Testing and Interpreting Interactions* Sage, Newbury Park, CA, 1992.
- Ancona, D.G., Goodman, P.S., Lawrence, B.S., and Tushman, M.L. "Time: A New Research Lens," *Academy of Management Review* (26:4) 2001, pp 645-663.
- Applegate, L.M., Austin, R.D., and McFarlan, F.W. *Corporate Information Strategy and Management* Harvard Business School Press, Boston, MA, 2007.
- Athey, S., and Schmutzler, A. "Product and Process Flexibility in An Innovative Environment," *Rand Journal of Economics* (26:4) 1995, pp 557-574.
- Austin, R., and Devin, L. *Artful Making: What Managers Need to Know About How Artists Work* Financial Times Prentice Hall, 2003.
- Beck, K., and Andres, C. *Extreme Programming Explained: Embrace Change*, (2nd ed.) Addison-Wesley, Boston, MA, 2005.
- Belsley, D.A., Kuh, E., and Welsch, R.E. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity* Wiley, New York, NY, 1980.
- Berry, D., Hungate, C., and Temple, T. "Delivering Expected Value to Users and Stakeholders with User Engineering," *IEEE Software* (42:4) 2003, pp 542-567.
- Boehm, B.W., and Turner, R. *Balancing Agility and Discipline: A Guide for the Perplexed* Addison-Wesley, 2004.
- 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, pp 279-306.
- Cockburn, A. *Agile Software Development* Addison-Wesley, Boston, MA, 2001.
- Cockburn, A. *Agile Software Development: The Cooperative Game*, (The Second Edition ed.) Addison-Wesley, 2007.
- Cramton, C.D. "The Mutual Knowledge Problem and its Consequences for Dispersed Collaboration," *Organization Science* (12:3) 2001, pp 346-371.
- Curtis, B., Kellner, M., and Over, J. "Process Modeling," *Communications of the ACM* (35:9) 1992, pp 75-90.
- Cusumano, M.A. "Managing Software Development in Globally Distributed Teams," *Communications of the ACM* (51:2) 2008, pp 15-17.
- 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, pp 135-169.

- Fitzgerald, B. "Systems Development Methodologies: The Problem of Tenses," *Information Technology and People* (13) 2000, pp 13-22.
- Floyd, S., and Lane, P. "Strategizing Throughout the Organization: Managing Role Conflict in Strategic Renewal," *Academy of Management Review* (25) 2000, pp 154-177.
- Gibson, C.B., and Birkinshaw, J. "The Antecedents, Consequences, and Mediating Role of Organizational Ambidexterity," *Academy of Management Journal* (47:2) 2004, pp 209-226.
- Gupta, A.K., Smith, K.G., and Shalley, C.E. "The Interplay between Exploration and Exploitation," *Academy of Management Journal* (49:4), Aug 2006, pp 693-706.
- Harter, D.E., Krishnan, M.S., and Slaughter, S.A. "Effects of Process Maturity on Quality, Cycle Time, and Effort in Software Product Development," *Management Science* (46:4) 2000, pp 451-466.
- 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, pp 481-494.
- Humphrey, W.S. *Managing the Software Process* Addison-Wesley, 1989.
- Humphrey, W.S., Snyder, T.R., and Willis, R.R. "Software Process Improvement at Hughes Aircraft," *IEEE Software* (8:4) 1991, pp 11-23.
- Im, G., and Rai, A. "Knowledge Sharing Ambidexterity in Long-Term Interorganizational Relationships," *Management Science* (54:7) 2008, pp 1281-1296.
- Jalote, P. *CMM in Practice: Processes for Executing Software Projects at Infosys* Addison-Wesley, Boston, MA, 2000.
- Klimoski, R.J., and Mohammed, S. "Team Mental Model: Construct or Metaphor," *Journal of Management* (20:2) 1994, pp 403-437.
- Krishnan, M.S., and Kellner, M.I. "Measuring Process Consistency: Implications for Reducing Software Defects," *IEEE Transactions on Software Engineering* (25:6) 1999, pp 800-815.
- Krishnan, M.S., Kriebal, C.H., Kekre, S., and Mukhopadhyay, T. "An Empirical Analysis of Productivity and Quality in Software Products," *Management Science* (46:6) 2000, pp 745-759.
- 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, pp 313-327.
- 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, pp 35-40.
- 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, pp 75-92.
- 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, pp 87-114.
- Levinthal, D.A., and March, J.G. "Thy Myopia of Learning," *Strategic Management Journal* (14) 1993, pp 95-112.
- Lyytinen, K., and Rose, G.M. "Information System Development Agility as Organizational Learning," *European Journal of Information Systems* (15:2) 2006, pp 183-199.
- 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, pp 1865-1883.
- 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, pp 273-283.
- Nidumolu, S.R. "Standardization, Requirements Uncertainty and Software Project Performance," *Information & Management* (31) 1996, pp 135-150.
- 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, pp 593-616.
- Podsakoff, P.M., and Organ, D.W. "Self-Reports in Organizational Research: Problems and Prospects," *Journal of Management* (12:4) 1986, pp 531-544.
- Pyster, A.B., and Thayer, R.H. "Software Engineering Project Management 20 Years Later," *IEEE Software*:September/October) 2005, pp 18-21.
- Ramesh, B., Cao, L., Mohan, K., and Xu, P. "Can Distributed Software Development Be Agile?," *Communications of the ACM* (49:10) 2006, pp 41-46.
- Royce, W. "Successful Software Management Style: Steering and Balance," *IEEE Software*:September/October) 2005, pp 40-47.
- Sambamurthy, V., and Kirsch, L.J. "An Integrative Framework of the Information Systems Development Process," *Decision Sciences* (31:2) 2000, pp 391-411.
- Standish Group "The Chaos Report," The Standish Group.

- Tiwana, A., Bharadwaj, A., and Sambamurthy, V. "The Antecedents of Information Systems Development Capability in Firms: A Knowledge Integration Perspective," International Conference of Information Systems, Seattle, WA, 2003, pp. 246-258.
- Tushman, M.L., and O'Reilly, C.A. "Ambidextrous Organizations: Managing Evolutionary and Revolutionary Change," *California Management Review* (38:4) 1996, pp 8-30.
- van Oosterhout, M., Waarts, E., and van Hillegersberg, J. "Change Factors Requiring Agility and Implications for IT," *European Journal of Information Systems* (15) 2006, pp 132-145.
- Xia, W., and Lee, G. "Complexity of Information Systems Development Projects: Conceptualization and Measurement Development," *Journal of Management Information Systems* (22:1) 2005, pp 45-83.