

12-27-2013

Developing the Agile IS Development Practices in Large-Scale IT Projects: The Trust-Mediated Organizational Controls and IT Project Team Capabilities Perspectives

Jenson Chong-Leng Goh
SIM University, jensongohcl@unisim.edu.sg

Shan L. Pan
National University of Singapore, University of New South Wales, pansl@nus.edu.sg

Meiyun Zuo
Renmin University of China, zuomy@ruc.edu.cn

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

Recommended Citation

Goh, Jenson Chong-Leng; Pan, Shan L.; and Zuo, Meiyun (2013) "Developing the Agile IS Development Practices in Large-Scale IT Projects: The Trust-Mediated Organizational Controls and IT Project Team Capabilities Perspectives," *Journal of the Association for Information Systems*, 14(12), .
DOI: 10.17705/1jais.00348
Available at: <https://aisel.aisnet.org/jais/vol14/iss12/1>

This material is brought to you by the AIS Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in *Journal of the Association for Information Systems* by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Journal of the Association for Information Systems

JAIS 

Research Article

Developing the Agile IS Development Practices in Large-Scale IT Projects: The Trust-Mediated Organizational Controls and IT Project Team Capabilities Perspectives

Jenson Chong-Leng Goh
SIM University
jensongohcl@unisim.edu.sg

Shan L. Pan
National University of Singapore
University of New South Wales
pansl@nus.edu.sg

Meiyun Zuo
Renmin University of China
zuomy@ruc.edu.cn

Abstract

This paper is based on multiple case studies on the construction of the Beijing Capital International Airport Terminal 3 in preparation for the 2008 Olympic Games that investigated the processes of instilling agile IS development practices in large-scale IT projects. This study develops useful theoretical constructs that will help researchers and practitioners who wish to learn about agile IS development practices as developed in large-scale IT projects. Adopting a contingent view, we uncover four factors that are critical in this development processes, namely: project uncertainty profile and project completion urgency; IT project team capabilities; organizational control mechanisms; and trust relationships among the IT project team, the vendors, and the users. Depending on the unpredictable nature of the project and the trust level among the IT project team, the vendors, and the users, we have uncovered the IT project team capabilities and the organizational control mechanisms that are needed to assure the success of a large-scale IT project. We posit that the interplay between the IT project team capabilities and the trust-mediated organizational control mechanisms forms the theoretical basis that defines agile IS development practice in large-scale IT projects. We argue that our findings provide insights to practitioners who are attempting to introduce agile IS development practices into any large-scale IT project. From a research perspective, the theory developed in this paper also sheds light on the importance of adopting a contingency view when researching agile IS development practices in a large-scale IT project and the factors to consider. This underpinning theoretical perspective will aid in the design of all future researches.

Keywords: Project Management, Organizational Control, Trust, Agile ISD practice, ISD Agility, Case Study.

* Gary Klein was the accepting senior editor. This article was submitted on January 26, 2012 and went through three revisions.

Volume 14, Issue 12, pp. 722-756 December 2013

Developing the Agile IS Development Practices in Large-Scale IT Projects: The Trust-Mediated Organizational Controls and IT Project Team Capabilities Perspectives

1. Introduction

Many practitioners and researchers in today's dynamic business environment find difficulty in achieving success in their IT project implementations. Despite years of research on IT project management and despite the wide availability of practical knowledge on project management, many IT projects still fail (The Standish Group International, 2009). According to the *CHAOS Summary Report*, which has tracked IT project failure rates since 1994, only 32 percent of the IT projects investigated was successful in 2009 (The Standish Group International, 2009). The same report also presented a worrying trend since 2002 of an increased IT project failure rate: from 15 percent to 24 percent. With increasing time-to-market pressure and an unprecedented rate of change in the user requirements during the system development in recent times (Lee, Venkatraman, Tanriverdi, & Iyer, 2010; Pacheco-de-Almeida, 2010), implementing IT projects successfully is becoming increasingly challenging even for a highly experienced software team (Geneca, 2011; Schmidt, Lyytinen, Keil, & Cule, 2001). This is apparent to the extent that more than 75 percent of the IT professionals surveyed deemed their IT projects as "doomed" right from the start (Geneca, 2011). As a result, the practitioner and research communities largely agree that the current state of IS development practices needs to be improved (Dingsøyr, Nerur, Balijepally, & Moe, 2012).

Since 2001, many scholars and practitioners have regarded agile IS development as the solution to the high rate of IT project failures (Abrahamsson, Conboy, & Wang, 2009). Advocating a close developer-user collaboration, agile IS development practices allow any development team to constantly adapt its developmental efforts toward ever-changing business requirements (Lee & Xia, 2010). Many scholars have deemed a close developer-user partnering process as the critical success factor in the increasingly volatile IT project development environment (Hsu, Liang, Wu, Klein, & Jiang, 2011; Jiang, Klein, & Chen, 2006; Wang, Shih, Jiang, & Klein, 2006). Not surprisingly, popular agile IS development practices that embrace such collaboration processes between developers and users have often enjoyed higher chances of project success (Stephens & Rosenberg, 2003). Many organizations regard these practices as a better collective solution to improve the quality of user interaction, user satisfaction, and project management processes (Ceschi, Sillit, Giancarlo, & De Panfil, 2005) compared to traditional IS development practices. As a result, many organizations have adopted agile IS development practices into their mainstream application development processes over the last decade (Finley, Wilson, & van Huizen, 2012). Unfortunately, extant research on the agile IS development has not been able to catch up with this increased demand for knowledge on the adoption of these practices. The emergence of several IS journals' special issues that have covered agile IS development practices supports this conclusion (e.g., Abrahamsson et al., 2009; Ågerfalk, Fitzgerald, & Slaughter, 2009; Dingsøyr et al., 2012; Erickson, Lyytinen, & Siau, 2005).

Despite the large number of empirical studies that investigate agile IS development practices in realistic contexts (e.g. Fitzgerald, Hartnett, & Conboy, 2006; Grenning, 2001; Mangalaraj, Mahapatra, & Nerur, 2009; Murru, Deias, & Mugheddu, 2003), we identify two gaps in the literature of these practices. The first gap concerns whether agile IS development practices can be developed in large-scale IT projects (Dingsøyr et al., 2012; Freudenberg & Sharp, 2010). Based on a survey conducted during the 11th International Conference on Agile Software Development in 2010, 300 practitioners identified the most-desirable research topic as "the adoption of agile IS development practices in large-scale IT projects" (Freudenberg & Sharp, 2010); and, in the same survey conducted in 2011, academics identified the same topic (Dingsøyr et al., 2012). Although scholars have researched the application of agile IS development practices in large IT projects to some extent (e.g., Elshamy & Elssamadisy, 2006), our current understanding of them is still very limited (Abrahamsson et al., 2009; Dingsøyr et al., 2012).

The second gap, according to several comprehensive reviews of the agile IS development literature (e.g., Abrahamsson et al., 2009; Dingsøyr et al., 2012; Dybå & Dingsøyr, 2008), is that "not enough attention [has been] paid to establishing theoretical underpinnings when investigating agile

development and its various practices". Because agile IS development is a research area that is primarily practitioner driven, and because of the lack of sound theoretical ground to explain agile IS development's essential methodology-independent concepts (Abrahamsson et al., 2009; Jacobson & Spence, 2009), accumulating knowledge on the adoption and application of agile IS development practices has been a challenging process (Abrahamsson et al., 2009; Erickson et al., 2005). This lack of theoretical foundation has led to much confusion for those who practice and research these practices (Conboy, 2009). As a result, many researchers have realized the importance of establishing a robust theoretical basis to guide ongoing research (Abrahamsson et al., 2009; Ågerfalk et al., 2009; Dingsøy et al., 2012).

In this study, we address these two gaps in the extant agile IS development literature. By conducting a case study based on four large-scale IT projects used to construct Beijing Capital International Airport (BCIA) Terminal 3, we address the knowledge gap of the agile IS development practices in large-scale IT projects, and leverage the literature of trust-mediated organizational control and IT project team capabilities to develop the theoretical underpinnings that outline the essential ingredients of an agile IS development practice in a large-scale IT project. In so doing, we provide the essential methodology-independent concepts that can serve as powerful theoretical glue to bind all future agile IS development research on large-scale IT projects (Jacobson & Spence, 2009; Whetten, 1989). Our research question is: How are agile IS development practices developed in a large-scale IT project?

2. Theoretical Background

2.1. Agile IS Development Practices

2.1.2. Introduction

Since the publication of the *Agile manifesto* in 2001 (Beck et al., 2001), agile IS development practices have received significant interest from many practitioners. Various forms of agile IS development practices (such as eXtreme Programming and Scrum) that are, based on the principles developed in the *Agile Manifesto*, have emerged. Practitioner-initiated research, inspired by many of those practices, has also created significant interest in the research community, (Dingsøy et al., 2012; Dybå & Dingsøy, 2008). Notwithstanding the efforts of pioneer research, we have subsequently identified two gaps in the research on agile IS development; namely, whether agile practices can be extended to large-scale IT projects, and the lack of theoretical underpinnings in research on agile IS development.

2.1.2. Extending Agile IS Development to Large-Scale IT Projects

Many new age practitioner-driven agile IS development practices have been reported to be highly effective in achieving project success in many small-scale or low-risk IT projects (Highsmith & Cockburn, 2001; Wang, 2007). In the face of the high dynamism in today's competitive business environment, both practitioners and researchers increasing want to extend these successful agile IS development practices to the large-scale IT projects (Dingsøy et al., 2012; Freudenberg & Sharp, 2010). Unfortunately, extending the agile IS development practices successfully adopted in small-scale IT projects directly into large-scale ones is not so straightforward (Cohen, Lindvall, & Costa, 2004).

A large IT project often involves a huge number of cross-departmental and/or cross-organizational team members (such as IT staff, users, customers, and IT vendors). As such, the technical domain knowledge required for a project's success are often thinly spread across the project's stakeholders (Curtis, Krasner, & Iscoe, 1988). Therefore, IT teams' management capabilities, such as dealing with the fluctuating and conflicting system requirements and avoiding communication and coordination breakdowns among team members (Curtis et al., 1988) are highly critical. These contemporary agile IS development practices, which are optimally designed for the small co-located team, are contrived when applied to large IT projects involving cross-departmental or cross-organizational team members. In addition, due to ever-evolving scopes and unpredictable development schedules when implementing these contemporary agile IS development practices, organisations face challenges in

planning and justifying the use of their IT resources to implement these practices in large IT projects (Boehm, 2002; Highsmith & Cockburn, 2001).

Some researchers have argued that the agile IS development practices cannot be extended to any large IT project (e.g., Rasmusson, 2003). More conventionally, other researchers have adopted a contingency view on the adoption of agile IS development practices in small IT projects, and hinted at the potential of using this view to extend agile IS development practices to large projects. For instance, some researchers state that organisations should customize agile IS development practices based on varying factors, such as IS development outcomes (Cao, Kannan, Xu, & Ramesh, 2009), the management of IS development knowledge (Nerur, Mahapatra, & Mangalaraj, 2005), and the on-the-job learning through experimentation and introspection (Nerur & Balijepally, 2007). Nonetheless, little research that explains how agile IS development practices can be fully embraced in large IT projects exist at present (Dingsøyr et al., 2012; Dybå & Dingsøyr, 2009; Erickson et al., 2005).

2.1.3. Lack of Theoretical Basis in Agile IS Research

The core values, principles, and practices of contemporary agile IS development have been developed based “mainly [on] past experiences and its effectiveness has been supported largely by anecdotal evidence and rhetorical arguments” (Lee & Xia, 2010, p. 88). Consequently, contemporary agile IS development practices suffer from a poor theoretical basis to support their future adoption and development (Dingsøyr et al., 2012). For instance, Lee and Xia (2010) argue that the lack of such theoretical underpinning in contemporary agile IS development practices has resulted in many organizations adopting these “practices without clearly understanding how agility is defined and measured and what factors they can control to influence it” (p. 88). Their viewpoint is also supported by several other researchers (e.g., Conboy, 2009; Dingsøyr et al., 2012) and other renowned agile practitioners (e.g., Jacobson & Spence, 2009). Indeed, both scholars and practitioners have called for more focused approach to theorizing about agile IS development. For instance, various definitions for what constitutes agility in IS development practice complicates efforts to understand the extant agile IS development literature (Conboy, 2009). In other words, in a situation where agility’s definition differs, various conclusions could be drawn from a single research and so lead to confusing and conflicting results. Such a situation harms the credibility of research on agile IS development (Abrahamsson et al., 2009). Dingsøyr et al. (2012) point out the importance of theory-driven (in contrast to practitioner-driven) agile IS development research: this research helps scholars and practitioners to differentiate true innovations from reinventions and remixes of old agile IS development practices, which, in turn, helps accelerate organizations’ future adoption of such innovations.

2.1.4. Building Theories on Two Existing Theoretical Foundations

Our study develops a theory that is built on existing theoretical foundations. One important foundation that our theory builds on is Conboy’s (2009) taxonomy for what constitutes agility in IS development practice (see Table 1). In short, Conboy (2009)’s taxonomy is a “theoretical validity tool” that helps to determine if the agile IS development practices applied in the large-scale IT project developed in our study are indeed agile.

The second theoretical foundation is the contingency view, which defines the success of agile IS development practices in small-scale IT projects (e.g., Cao et al., 2009). We posit that two contingencies will greatly influence the success of a large-scale IT project: (1) the uncertainties in the project’s requirements, and (2) the high urgency to complete the project (Meyer, Loch, & Pich, 2002). For instance, if few uncertainties surface in a project’s requirements, changes will be minimal during the IS development and, hence, traditional plan-driven IS development practices would be more appropriate than agile IS development practices (Austin & Devin, 2003; Harris, Collins, & Hevner, 2009). Moreover, with no urgency to complete a project, there is little need for any IS development to be agile because the IT project team can take its time to incorporate changes into the business requirements during a system implementation. Therefore, we argue that these two contingencies necessitate the development of agile IS development practices in large-scale IT projects. Thus, these two contingencies lead us to question what the key ingredients of agile IS development practices are if they are to be effectively adopted into large-scale IT projects.

Table 1. Taxonomy of the Agile IS Development Practice (Conboy, 2009)

1. To be agile, an IS development method component* must contribute to one or more of the following: (a) creation of change, (b) proaction in advance of change, (c) reaction to change, and (d) learning from change
2. To be agile, an IS development method component must contribute to one or more of the following (and must not perform poorly in any one of them): (a) perceived economy, (b) perceived quality, and (c) perceived simplicity
3. To be agile, an IS development method component must be continually ready (i.e., requiring minimal time and cost to prepare the components for use)

* An IS development method component refers to any distinct part of the IS method.

2.2. Trust-Mediated Organizational Control

We premise the use of organizational control as one of the key ingredients in an agile IS development practice for any large-scale IT project. Appropriating organizational control as one of the key ingredients is essential because IS development practices have been defined as a set of socially defined ways in system development that defines outcomes and creates the basis for responding appropriately to individual circumstances (Ashurst, Doherty, & Peppard, 2008). Inevitably, “a set of socially defined ways” requires the application of organizational control mechanisms by the project managers (a.k.a. controllers) to ensure that their direct and/or indirect reports (a.k.a. controllees) conforms to the “defined ways” during the system implementation (Kirsch, 1997). Extant literature on organizational control has generally converged in defining organizational control as “encompassing all attempts to ensure that individuals in an organization act in a manner that is consistent with meeting the organization’s goals and objectives” (Eisenhardt, 1985; Kirsch, 1997; Ouchi, 1980). Kirsch (1997)’s definitions of organizational control have been widely established and adopted as the de-facto models of choice whenever organizational control is discussed, especially in studies about IS development (see Table 2).

Table 2. Definition of Control Mode (Kirsch, 1997)

	Types	Definition
Formal control	Behavior control	The act of setting a set of specific rules and procedures that aligns the behavior of the controllee to whatever is deemed desirable in meeting the organization’s objectives.
	Outcome control	The act of setting goals that are desirable in meeting the organization’s objectives and rewards are given to any controllee who meets them.
Informal control	Clan control	Looks at the dissemination of desired values, beliefs, and philosophies of the organization by influencing the individuals to a common set of norms and values within the clan.
	Self control	Looks at mechanisms that identify and provide a conducive environment that rewards and encourages highly motivated individuals or groups of individual to exercise self-control in the best interest of meeting the organization’s objectives.

The agile IS development literature also contains several empirical studies that analyze organizational control theory’s effects on the quality of agile IS development practices (e.g., Harris et al., 2009; Maruping, Venkatesh, & Agarwal, 2009). However, we have two key concerns with the way organizational control theory is being used in the existing agile IS development literature. Firstly, we note that organizational control theory is often applied without much consideration for the trust relationships among IT project teams, outsourced vendors, and users. Yet, the effectiveness of an organizational control mechanism is often dependent on the trust relationships between IT project teams and outsourced vendors (Inkpen & Currall, 2004) and between the IT project team and the

users (Wang, Ju, Jiang, & Klein, 2008). The trust relationships among an IT project team and its stakeholders also influence the types of organizational control mechanisms that would be applied in an IT project (Rustagi, King, & Kirsch, 2008). Furthermore, in the context of a large-scale IT project, controlling outsourced vendors (Kirsch, Ko, & Haney, 2010) and users (Jiang et al., 2006) is a particularly common and salient activity that is critical to success. Hence, we surmise that trust-mediated organizational control is a more-appropriate choice in the context of our study.

Secondly, it is problematic that these empirical studies adopt the use of the practitioner-driven agile IS development practices as their foundation (e.g., Maruping et al., 2009). Because these practitioner-driven agile IS development practices interpret the agile practices in IS development projects differently (Lee & Xia, 2010), studies made using it as a foundation would find it challenging to extend their findings to other contexts (e.g., to large-scale IT projects). This is one of the reasons why some researchers are calling for greater care in the use of practitioner-driven agile IS development practices in research (e.g., Lee & Xia, 2010), while others are strongly encouraging the research community to develop agile IS development theories drawn from theory found in the literature (e.g., Conboy, 2009; Dingsøyr et al., 2012). As such, we chose to develop the theoretical models in our study using the theory of trust-mediated organizational control as one of our theoretical foundations.

2.3. IT Project Team Capabilities

IT project team capabilities have long been regarded as the essence of organizational excellence and IT project success. For instance, prior studies on IT project team capabilities have shown that an IT project team's ability to engage users actively and effectively can lead to IT innovativeness (Kettinger & Lee, 2002) and IT project success (Hsu et al., 2011). Appropriately applying an IT project team's technical mastery in combination with the business expertise from users has been shown to provide the best environment for deriving new IT knowledge that is crucial to the IT project success (Nambisan, Agarwal, & Tanniru, 1999). However, in order for an IT project team to reap such benefits, they must possess a blend of technical, behavioral, and business knowledge and skills (Bassellier & Benbasat, 2004; Fink & Neumann, 2007). Without the development of these IT project team capabilities, efforts to redesign the business processes of a large-scale IT project will be seriously impaired, and the project will most likely end in failure (Rockart, Earl, & Ross, 1996; Ross, Beath, & Goodhue, 1996).

According to Fink and Neumann (2007), the literature on the IT project team capabilities can be broadly divided into: (1) technical capability, (2) behavioral capability, (3) business capability, and (4) infrastructure capability. We adopt this categorisation and develop a list of IT project team capabilities that we regard as critical to successfully implement an IT project (see Table 3).

2.4. Relationships Among Project Uncertainty, Project Completion Urgency, Trust-Mediated Organizational Control, and IT Project Team Capabilities

When projecting the success of a large-scale IS development, there is always a perceived risk often derived from uncertainties about the project's requirements (Harris et al., 2009; Meyer et al., 2002) and the urgency to complete the project. Thus, the project team is required to develop an ability to sense-and-respond to this perceived risk (Fink & Neumann, 2007; Lee & Xia, 2010). The availability of these capabilities in an IT project team determines the appropriate application of organizational control mechanisms during the IS development (Kirsch et al., 2010), and the application of organizational control mechanisms can, in turn, lead to the development of new IT project team capabilities that contribute positively to an IS project's success (Kirsch, 2004; Kohli & Kettinger, 2004). For instance, on the one hand, if the IT project team is capable of establishing clarity in the tasks and project outcomes, it becomes a key antecedent for selecting organizational control mechanisms during the IS development (Kirsch, 2004; Kirsch et al., 2010). On the other hand, applications of organizational control mechanisms that are done well enable the derivation of an IT project team's capabilities, which ensures the success of the IS development (Kirsch, 2004; Kohli & Kettinger, 2004). To summarize, based on our literature review, we deduce that trust-mediated organizational control mechanisms and IT project team capabilities are the two "key ingredients" necessary to effectively address the elements of uncertainty in projects and the urgency for their completion when adopting agile IS development practices.

Table 3. List of the IT Project Team Capabilities Critical to a Project's Success Based on Fink & Neumann's (2007) Categorization

Category	Constructs and references
Technical capability: the ability of the IT personnel based on their specific expertise in technical areas	Technology change and growth management (Mata, Fuerst, & Barney, 1995) IT change management (Bharadwaj, 2000) IT HR asset management (Ross et al., 1996) IT/Innovation Governance and IT project management (Tarafdar & Gordon, 2007)
Behavioral capability: the interpersonal and management ability of the IT personnel interacting with and managing others	Stakeholder management (Wade & Hulland, 2004) IT change management (Bharadwaj, 2000) Absorptive capacity (Cohen & Levinthal, 1990) Shared ideology creation capability (van Den Bosch, Volberda, & De Boer, 1999) Knowledge integration (Ayas & Zeniuk, 2001; Boland & Tenkasi, 1995) Business-IS linkages (Tarafdar & Gordon, 2007) Quality of IT business expertise and relationship infrastructure (a.k.a. competitive capability) (Bhatt & Grover, 2005) Business competence of IT professional (Bassellier & Benbasat, 2004) IT-business partnering capability (Wang et al., 2006)
Business capability: the ability of the IT personnel to understand the overall business environment and the specific organizational context	Absorptive capacity (Cohen & Levinthal, 1990) Knowledge integration (Ayas & Zeniuk, 2001; Boland & Tenkasi, 1995) Technology and relationship asset management (Ross et al., 1996) Knowledge management (Tarafdar & Gordon, 2007) Intensity of organizational learning a.k.a. dynamic capability (Bhatt & Grover, 2005) Business competence of IT professional (Bassellier & Benbasat, 2004)
Infrastructure capability: the ability of the IT unit to provide extensive firm-wide IT infrastructure services that support the organization's business process	Effective IT platforms management (Feeny & Willcocks, 1998) Capability reconfiguration (Lavie, 2006) IT infrastructure services capability (Weill, Subramani, & Broadbent, 2002) Quality of IT infrastructure a.k.a value capability (Bhatt & Grover, 2005)

3. Research Methodology

3.1. Method Selection and Research Setting

This study adopts the case study research methodology for two reasons. First, the case study research methodology can address concerns about the way to develop agile IS development practices in a large-scale IT project (Walsham, 1995). Second, it can uncover the complex relationships among project uncertainty and completion urgency, trust-mediated organizational control mechanisms, and the IT project team capabilities. This enables us to develop a better understanding on how these relationships contribute to form agile IS development practices in a large-scale IT project. Thus, the case study method is the most appropriate method to achieve our research's objective (Pentland, 1999).

Specifically, this study uses the large-scale project of building Beijing Capital International Airport Terminal 3 as its case study. We selected this project for the following reasons: (1) Terminal 3's (deemed as the world's largest Airport Terminal in 2008) IT implementation, consisting of many large-scale IT systems, provides an opportunity to look into "project completion urgency" because Terminal 3 needed to be ready in an ambitious deadline of around two years (in time for the 2008 Olympics), (2) it provides an opportunity to look into "project uncertainty" because this construction's large-scale IT projects were highly complicated and required close coordination among multiple stakeholders in order to achieve success. More importantly, the overall project turned out to be successful (Beijing Capital International Airport, 2009), and, hence, it demonstrates a case-in-point where uncertainty and completion urgency were successfully mitigated.

3.2. Data Collection

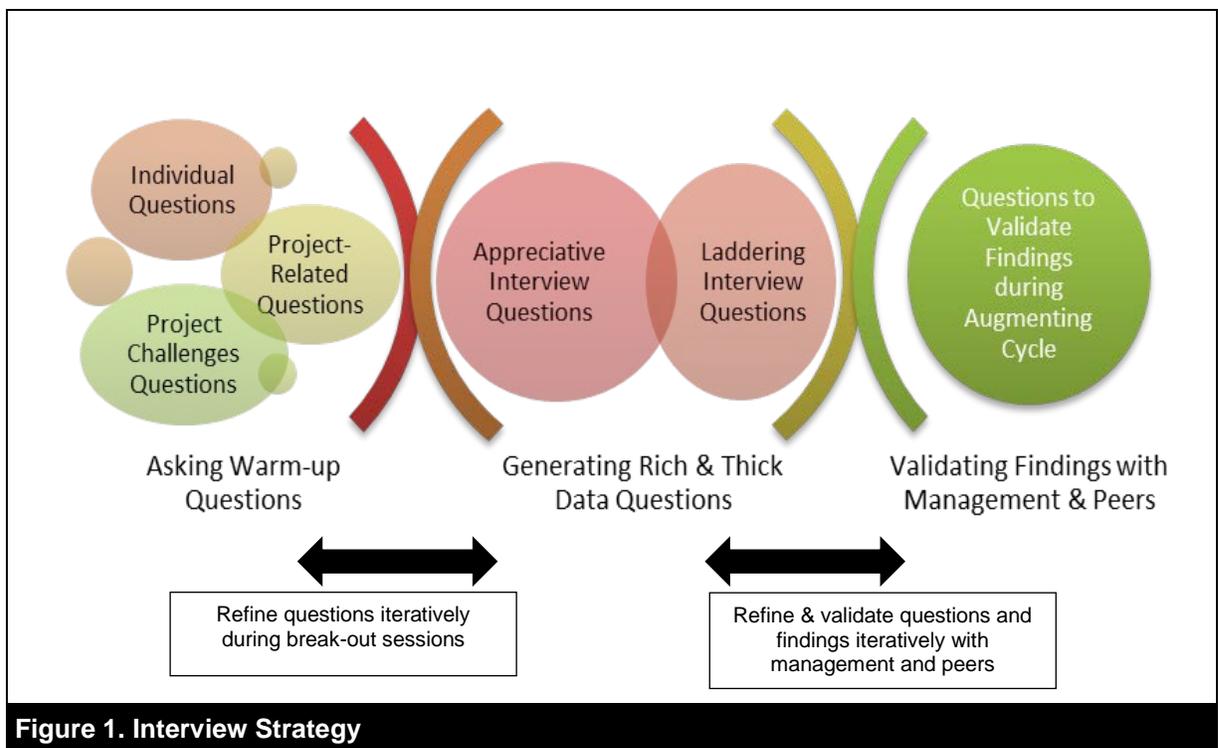
Access to the case site was negotiated and gained in June 2009. Subsequently, the study adopted a multiple case design, whereby six major IT systems and two terminal command centers that were crucial to the success of the Terminal's IT project were identified. The unit of analysis was at the project level. In total, interviews involving 27 individuals were conducted (see Table 4). Several key project stakeholders were interviewed multiple times to validate and ensure the reliability and validity of the data collected from these interviews. The duration of each interview averaged about 120-180 minutes. The interviews were digitally recorded and transcribed for subsequent data analysis. All interviewees were carefully selected based on the critical roles that they played in the IT projects. In each large-scale IT project chosen for the study, we identified the dominant IT project team's capabilities and dominant trust-mediated control mechanisms.

In our case study, we used four out of the six large-scale systems. These four systems were selected based on the magnitude of their implementation, their complexity in dealing with large groups of internal and external stakeholders, their unique nature of project uncertainty, and their criticality to the whole Terminal 3 project. The remaining two systems were dropped because many of the findings overlapped with the other systems featured in our paper, and therefore we reached the state of "theoretical saturation" (Eisenhardt, 1989).

As the number of interviewees in some of the large-scale IT projects was small, we used two interviewing techniques similar to those advocated by Schultze and Avital (2011) (namely, the appreciative and the laddering interviewing techniques) to generate rich descriptions from these small number of participants (see Figure 1 for details of our interviewing strategy). To ensure that the richness of the data captured from the interviews was properly validated, we rigorously cross-checked it on multiple occasions with various members of the IT department and management team. To further strengthen the reliability and validity of this data, a large amount of secondary data, comprising of project documents, books, photos, and videos, was also collected during the on-site interview sessions.

Table 4. Details of Interviews Conducted

Description	Role of individual interviewed in the Terminal 3 IT project	Number of individual interviewed
Off-site preliminary interviews	General Manager, BCIA Management	1
Overview interview	Deputy Manager, BCIA Management Project Manager, Data Center System Project	2
AODB system project	Project Manager, First generation AODB administrator Business Process Lead, AODB System Project Technical Lead, AODB & Data Center System Project	3
Airport departure system project	Technical Lead, Departure System Project Business Process Lead, Departure System Project	2
Airport security system project	Project Manager, Security System Project Technical Lead, Security System Project	2
Airport data center project	Deputy Manager, BCIA Management Project Manager, Data Center System Project Technical Lead, AODB & Data Center System Project	3
Airport flight display system project	Business Process Lead, Flight Display System Project Technical Lead, Flight Display System Project	2
Airport luggage system project	Deputy Manager, BCIA Management Project Manager, Luggage System Project Business Process Lead, Luggage System Project	3
Terminal & system command centers	Deputy Manager, BCIA Management Business Process Lead, Western SOCC Project Business Lead, AODB System Project	3
Final interviews and findings presentation	Executive Vice President cum CIO, BCIA Management Manager, BCIA Management, Deputy Manager, BCIA Management Project Manager, Data Center System Project Technical Lead, AODB & Data Center System Project Business Process Lead, Luggage System Project	6
Total		27



3.3. Coding and Data Analysis

We derived our case study design using Pan and Tan's (2011) structured-pragmatic-situational (SPS) approach as a guide (see Figure 2). Before we started the study, we derived a set of potential themes by thematically analysing the data obtained from the various secondary sources and from the preliminary offsite emails and phone interviews. With the help from our gatekeeper, we validated and confirmed a few identified themes that could be further investigated. This set of preliminary themes then served as a "sensitizing device" (Klein & Myers, 1999) that guided the subsequent interviewing process and the design of our interviewing strategy.

The on-site interviews were then conducted. Data analysis was performed concurrently with the data collection activities. To ensure our collected data aligned with our theoretical model, we specially designed break-out sessions in between interviews to discuss our findings, which allowed us to modify and refine the interview questions for subsequent interviews (Eisenhardt & Graebner, 2007). This data analysis work followed an iterative process of moving back and forth among empirical data, our theoretical lens, relevant literature, and the proposed model until we reached theoretical saturation as determined by the senior case study researchers in our team (Pan & Tan, 2011). To ensure the rigor of our findings, we presented them iteratively to the respective stakeholders at the Beijing Capital International Airport (BCIA)'s IT department to check their accuracy, correctness, and appropriateness (Walsham, 2006). After the on-site interviews, we conducted several rounds of confirmatory data analysis using the selective coding techniques (see Table 5) that took into account the new interview data and the large amount of project documents (such as the project schedule, the system design diagrams, etc) collected along with the new insights gained from the literature.

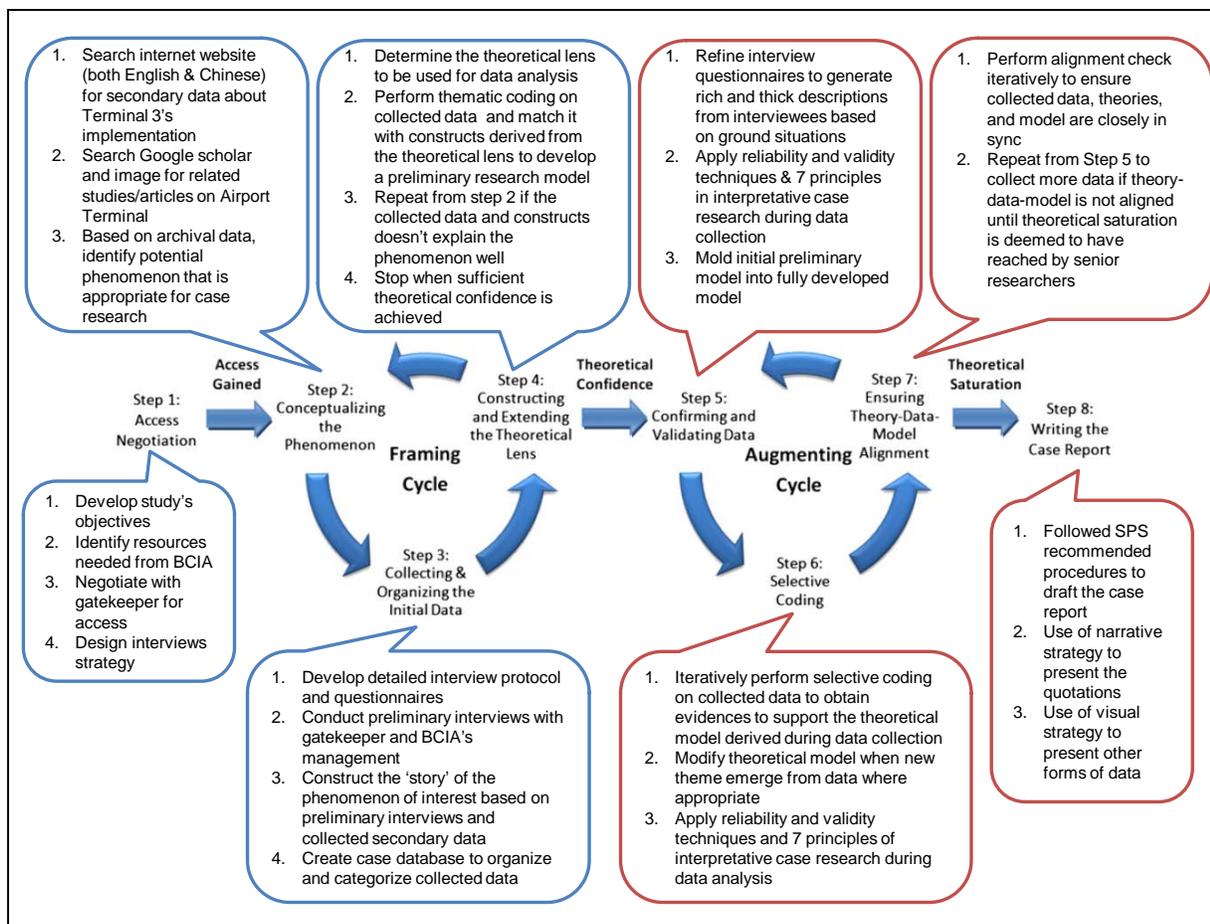


Figure 2. The Structured-Pragmatic-Situational (SPS) Approach to Case Study Design (Pan & Tan, 2001)

Table 5. Snippet of the Selective Coding Process on Airport Departure System's Case Data

Empirical observations (representative quotes from selective coding)	Theoretical observations	Theoretical construct
"If you have a chance to read the requirements submitted (for the departure system) by each airline, you will be surprised to realize how far into the future their predictions are for the airline industry ... and the needs of their passengers ... Because our customers are airlines, the requirements gathered from them are all very accurate in predicting the eventual use" (ADS#4)	IS Planning Capability	IT capabilities in the project team
"We provide training to the vendor. Through these training sessions, we transfer our company (BCIA)'s management methodology, procedures, norms, including our existing system operation details and things to look out for during the requirement study. We want them to meet our quality of work standards. Make their thought patterns, beliefs, work attitudes to be in sync with us" (ADS#6)	Behavioral Conditioning Capability	

The entire process of post data analysis took more than two years to complete. Telephone interviews were also conducted with the stakeholders in BCIA to maintain the accuracy of the derived theoretical models during this lengthy process. To ensure validity and reliability of our collected data, we adopted measures advocated in Figure 3 throughout our research.

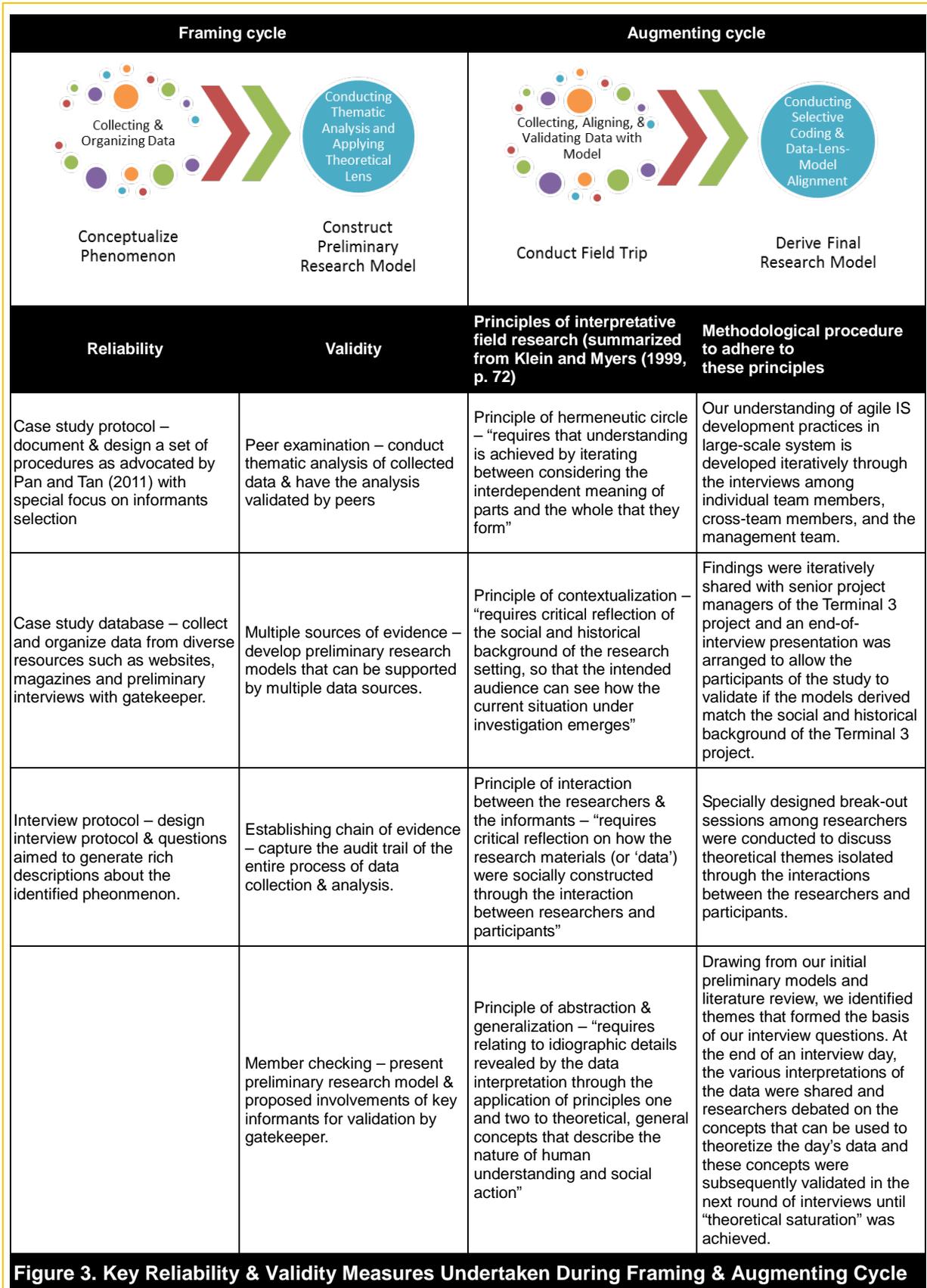


Figure 3. Key Reliability & Validity Measures Undertaken During Framing & Augmenting Cycle

Framing cycle		Augmenting cycle	
Reliability	Validity	Principles of interpretative field research (summarized from Klein and Myers, 1999, p. 72)	Methodological procedure to adhere to these principles
		Principle of dialogical reasoning – “requires sensitivity to possible contradictions between theoretical preconceptions guiding the research design and actual findings (the story which the data tell) with subsequent cycles of revision”	Comparing pre-study data with field data by multiple researchers twice daily allowed the contradictions between proposed theoretical preconceptions to be picked up and addressed.
		Principle of multiple interpretations – “requires sensitivity to possible differences in interpretations among the participants as are typically expressed in multiple narratives or stories of the same sequence of events under study; similar to multiple witness accounts even if all tell it as they saw it”	Multiple researchers were presented during all the interviews with clearly assigned roles (e.g. data recording, questioning, documenting observations). During break-out sessions and at the end of every interview day, these observations were shared and researchers debated before the observations were synthesized.
		Principle of suspicion – “requires sensitivity to possible biases and systematic distortions in the narratives collected from the participants”	Multiple data was collected from various sources to provide the ability to triangulate the correct sequence and consistency of events and facts. Findings in one system’s interviews were cross-validated with others where possible.

Figure 3. Key Reliability & Validity Measures Undertaken During Framing & Augmenting Cycle (cont.)

4. Case Description

4.1. Organizational Background

Beijing Capital International Airport (BCIA) Company Limited is a state-owned organisation that manages the main airport of Beijing, China. BCIA manages the Beijing Capital International Airport that consists of three airport terminals (Terminal 1, 2, and 3). As the capital airport of China, BCIA is constantly under intense pressure to cope with the rapid growth of air traffic to and from Beijing. In 1990, Terminal 1 was built to occupy a surface area of about 90,000 square meters. In the span of 10 years, Terminal 2 was constructed and has been in operation since 2000. It occupies a total floor space of 336,000 square meters. The colossal Terminal 3 was specially designed and constructed for the use of 2008 Olympic Games. It was built on a land area of 986,000 square meters—twice the size of Terminal 1 and 2 combined. The epic scale of the IT system implementation program for Terminal 3 was also unprecedented in BCIA’s history. In addition, BCIA was given the tall order to ensure the “zero incident, zero accident and zero complaint” mandate throughout the 2008 Olympic Games by the Chinese Government. Our case study focuses on BCIA’s IT department and how its IT project team mitigated the various project uncertainties inherent in four different large-scale IT projects.

4.2. Airport Departure System

The airport departure system's main functionality is to allow the airline departure systems to "hook up" in order to support the process of passengers checking in and to facilitate passenger boarding. The development work of the system for Terminal 3 started in 2006 and was completed in early 2008. The development work entailed many small customizations made to the system in order to meet the unique airline needs of serving passengers leaving the airport. Managing the other airlines' expectations and linking this system with the various airline departure systems seamlessly were the critical success factors for this project.

All services provided by the airlines at the counters are supplied by our Airport Departure System... from the perspective of the passengers, its smooth operation is of utmost importance ... because if it fails, passengers will not be able to leave [the airport] ... only small variations exist within this business process [of airport departure] across the airlines, the main business processes typically remain unchanged (ADS#1).

The IT project team faced some challenges when trying to ensure these critical success factors. For instance, as the propriety software was maintained overseas by its vendor, the speed of the implementation and the quality of the customizations were difficult to control internally. Also, because the members of the vendor's staff deployed to this project were inexperienced in handling the localized settings unique to this system, scheduling slips were a significant risk.

They sent two Australians to install their software ... and their technologies are proprietary ... Hence, I honestly feel that this is not a good approach because our system is localized [to China's requirements] ... regardless of which level of expertise they [the two Australians] possess, I don't think they can understand and resolve our [system] problems (ADS#2).

4.2.1. Mitigating the Challenges

The IT project team took the following steps to mitigate the risk of falling behind schedule. First, it closely controlled and monitored the scope and uncertainty specific to each customization for the airlines throughout the system development to minimize occurrences of rework. It was able to achieve this by appointing its in-house airport departure system expert as the project manager; and by convincing all the airlines to send in their subject domain experts from their respective airline departure systems to form a project steering committee.

I have been in Airport Departure System work for 9 years. Every detail [of the system] whether it is good or bad, I know it very well. We have the most professional people [refer to members of project steering committee which also includes the eventual users of the system] doing the professional work (ADS#3).

This project steering committee was responsible for the initial tender proposal and project planning; and, subsequently, the iterative project implementation and the change review processes. While changes to the requirements were evident during the system implementation, the project steering committee was instrumental in identifying the requirements for the customizations accurately in each change request, which reduced occurrences of needing any reworking during the system development.

If you have a chance to read the requirements submitted [for the departure system] by each airline, you will be surprised to realize how far into the future their predictions are for the airline industry ... and the needs of their passengers ... Because our customers are airlines, the requirements gathered from them are all very accurate in predicting the eventual use (ADS#4).

Second, BCIA had some reservations about the vendor's abilities to deliver the development work on schedule. Hence, the vendor was strictly managed to ensure the timely delivery of the system's requirements or users' change requests.

We are with the vendor all the time so much so that there is little chance that we will develop misunderstandings [in terms of system requirements]. Because we are seated together, if there is any problem that surfaces, we will immediately communicate and establish consensus on the spot (ADS#5).

A series of vendor-training sessions was conducted at the beginning of the project. These training sessions were designed to impart important information pertaining to the Beijing Capital International Airport's (BCIA) culture, practices, work attitudes, and standards to the vendor. The members of the staff employed by the vendor were obliged to align themselves to BCIA's culture, practices, work attitudes and, standards.

We provide training to the vendor. Through these training sessions, we transfer our company's [BCIA] management methodology, procedures, norms, including our existing system operation details and things to look out for during the requirement study. We want them to meet our quality of work standards. Make their thought patterns, beliefs, work attitudes to be in sync with us (ADS#6).

A rigorous and detailed "backward planning" (interpreted as the setting of hard deadlines and planning backward for milestones) methodology was also adopted during this system implementation. Weekly meetings were held to track the developmental progress of the vendor, and findings were presented to the project steering committee to ensure a constant alignment of deliverables with the identified system requirements and change requests.

Every week we have a project meeting to discuss the progress of the previous week (ADS#7).

We make use of a backward planning methodology. We look at our deadline, how many days it will take [before we reach it], then make milestones starting backwards (ADS#8).

The project manager even went to the extent of detailing a daily schedule of work activities for the vendor and deploying IT staff to ensure the conformance of the daily schedule.

The plans are very clear ... work details up to every day, some even up to every few hours. That is to say if the work on that day can't be completed, everyone will have to work overtime (ADS#9).

Leveraging the project steering committee's accurate identification of the user requirements in change requests, and the tight control over the progress of its vendor, the IT project team was able to keep the project on schedule amid a large number of requests for customization and change.

4.3. Airport Operation Database (AODB) System

The AODB system is a platform through which the entire set of airport terminal data is integrated and exchanged across all mission critical systems. The first custom-built AODB system was installed in Terminal 2 in 1999. When Terminal 1 was refurbished, the same custom-built AODB system was integrated into Terminal 1's operations, allowing one AODB system to manage the operations of the two terminals. The implementation of a new AODB system for Terminal 3 began in 2006 and was delivered in early 2008. During this project's tendering process, the IT department was aware of two integration challenges. First, it was challenging to install the new AODB in Terminal 3 because it was an off-the-shelf system that differed significantly from the custom-built AODB system used at Terminals 1 and 2.

This system is different from T2's [AODB system in Terminal 2] and is an off-the-shelf product ... because many of the business processes [of the AODB in Terminal 3] are different from Terminal 2 [AODB system] ... it [AODB system in Terminal 3] involves tight data integration with older systems [refer to the AODB system in Terminal 2] (AODB#1).

Second, all the AODB systems needed to be integrated with the airport traffic control (ATC) system managed by the Air Traffic Control Authority in Beijing. This system fed important flight and weather information to the AODB system that was undergoing a major upgrade, and added to the anxiety: "There are a lot of uncertainties on how to link up our AODB system with ATC during system development. This took a lot of our effort [to link the systems up]" (AODB#2).

4.3.1. Mitigating the Challenges

Given the great complexity and the unlimited possibility of use of the AODB system at the airport terminal, it was difficult for the management to explicitly define the outcome of the project. As a result, all AODB administrators in the IT department had to constantly experiment with the system. For instance, AODB#3 said: "our IT department has a very strong learning culture ... we invented many system maintenance strategies through constant experimentation", and AODB#4 said: "He [Head of IT department] encourages us to continuously do research and innovate ... we are very willing to do it and contribute our knowledge to make this industry better".

Over time, the efforts invested in this area resulted in three improvements. First, a foundation for developing a comprehensive AODB system administration and maintenance manual was set.

In 2000, I took the lead with a few other colleagues to look into an in-depth study on system backup and business continuity [for the AODB] ... during that time, we didn't leverage on outside help and we did all the research on our own ... From 2000 to 2002, we came up with a maintenance standard operation procedure handbook. Till today, our maintenance strategy is dependent on this handbook (AODB#5).

Second, two terminal command centers used for consolidating AODB-related systems and terminal operations for efficient and effective operational management were set up: "The setup of the two Terminal Command Centers consolidated and replaced all the old systems and processes [surrounding the maintenance and administration of AODB system]" (AODB#6).

Finally, tender requirements and requirements in change requests that were highly pre-emptive in mitigating potential uncertainties and risks in developing the AODB system at Terminal 3 were crafted. For instance, the IT project team was particularly proud of their ability to develop a highly flexible component in the Terminal 3 AODB system that considered all the possible changes in the upgrading exercise of the ATC system. As a result, they were highly effective in pre-empting changes in the ATC system that the AODB system needed to integrate with.

Because ATC is changing their system, when we complete our AODB project, they are still developing. Because of the experience gained from our constant experimentation, we are able to come up with a lot of contingency plans ... at the end, regardless of how they [ATC] change, we can easily accommodate our system to interface with theirs [ATC system] (AODB#7).

Because of our in-depth knowledge of T2's [Terminal 2] business operations, we can easily make a comparison between T2 and T3 [Terminal 3] and highlight the weaknesses and strengths of T2's system [AODB system], then we can use this knowledge to inform the vendors to improve their system by absorbing T2's strengths and eliminating its weaknesses (AODB#8).

The management of the IT department placed great trust in the hands of the highly experienced AODB administrators and the outsourced vendor. For instance, the project members and the vendor

were encouraged to explore all possible ways to deal with the inherent uncertainties about integrating the two AODB systems, and between the AODB systems and the airport transport control (ATC) system.

They [vendor] have very unique recommendations and understanding on system development ... during the project especially at the last stage, they can make predictions of the future maintenance model and preemptively did a lot of work (AODB#9).

We have learned a lot from them [vendor] including knowledge in system improvement, maintenance and various automations of system management processes (AODB#10).

At that time, the stress exerted on the department was the largest ... our most trusted and valuable staff [refer to the AODB Administrators who were also eventual users of the system] had left for the Terminal 3 project (AODB#11).

These staff [AODB administrators] possessed several rare attributes namely strong responsibility to work and being meticulous in nature ... we usually would identify a few and then develop them ... only the best get selected to be part of the core AODB administrator team (AODB#12).

After a high level of trust was built, several desirable behaviors in the members were perceived. For instance, they demonstrated selflessly shared knowledge and proactively anticipated problems during the project's implementation. The best example was the informal mentor-apprentice relationship, whereby the ex-AODB administrators (who were not directly involved in this project) served as mentors to the current AODB administrators in charge of this project.

We have an unofficial master-apprenticeship mechanism and a backup mechanism ... if the apprentice is good at work, the master will have the opportunity to do other things and can also be promoted. This serves as a big motivation for the master [to coach the apprentice] (AODB#13).

These key competences greatly contributed to the IT project team's ability to keep the project on schedule.

4.4. Airport Security System

The tender for this system was awarded in October 2005 and it went live in March 2008. As one of the most critical systems directly linked to the "zero incident, zero accident and zero complaint" objectives, the airport security system was of utmost importance. In order to meet this requirement, the most stringent security screening processes had to be mandated. While there existed security systems in Terminal 1 and 2, none was as complex or as large as the one at Terminal 3. Due to the sheer size of Terminal 3, the level of close collaboration among stakeholders was also unprecedented. This resulted in a lot of uncertainties in forecasting the system requirements during the tender preparation phase.

This security system is the first of its kind in China and you may not find any system of such a nature in the world ... requirements gathering is extremely challenging because it involves a large number of stakeholders and the system needs to integrate data from many data sources (ASS#1).

This project involves a lot of stakeholders ... including the immigration authority's various departments ... and the whole airline company's ground operations ... to develop this system, it requires the consolidation of knowledge from various sources to design the various system connecting points for these stakeholders ... gathering requirements [from these stakeholders] is a really difficult endeavor [in this project] (ASS#2).

4.4.1. Mitigating the Challenges

To mitigate these challenges, the IT project team spent a significant amount of their time visiting many terminals around the world to acquire updated knowledge about implementing airport security systems. The knowledge acquired from these visits was subsequently used in multiple feasibility studies. Coupled with the experience accumulated from maintaining and implementing the airport security system in Terminal 1 and 2, the tender requirements were finally drafted:

In April 2005, before the beginning of the tender, we did a number of visits and research studies on a number of airports ... we involved the design unit in T2 [Terminal 2] to consolidate our findings into the tender specification for the security system in T3 [Terminal 3] (ASS#3).

To ensure that a close collaboration among the stakeholders was achieved, the IT project team tried to instill a sense of pride in the project's stakeholders. They used multiple channels of communication (especially encouragement slogans) to motivate stakeholders to answer to the higher calling of their work.

With regard to work culture ... the slogan in our canteen can best represent that and it says "If you are afraid of death, don't become a communist. If you are afraid of hard work, don't take up system development work of T3!" (ASS#4).

These measures were effective to induce a shared sense of pride and privilege among the stakeholders who were proud to be associated with the "novelty" of the security system implementation.

Everyone who is participating in this project will have a revelation that this is the most political and critical project [in T3] (ASS#5).

This is a political project ... all people [partners, users and IT staffs] who participate in this project know how important this project is ... hence we have a collective target and even when there are conflicts we will find ways to address them collectively (ASS#6).

After the project implementation phase commenced, the IT project team developed several practices that promoted risk-taking and self-learning behaviors among the members. For instance, a scheme known as the "joint responsibility deposit" was created. The vendor, the users, and the team members were approached to make proposals on how to resolve an unforeseen problem that had occurred during the project implementation, and they had to "put their money where their proposals were" under this scheme. Conversely, anyone who made a proposal and was able to resolve the problem in a timely fashion would be returned the deposit and gained additional monetary rewards. However, if his or her proposal failed, the deposit would not be refunded.

If you achieve your target, you will be given a monetary reward, if you don't, money is deducted... once we established the target, everyone will come up with the "responsibility" deposit, you don't hit it [target] we deduct money from the deposit. You complete it, we give you a reward (ASS#7).

There was a prevailing low level of trust among all the stakeholders (the users and the vendor) in this project:

Because our system development schedule is fixed and their [users] business processes are only finalized at the tail end of the project, we cannot wait for them, we use the processes done within our system and get them to adopt it (ASS#8).

... we have a change management process that is very stringent. Any change [proposed by vendor/users] needs to go through the engineer in the audit team first. Only when there is sufficient evidence then it goes up to our change management committee for approval (ASS#9).

As such, an independent audit team was created to inspect all the project deliverables and to perform the first-cut checks on all the change requests before they were escalated to the change management committee. Without the audit team's endorsement, the subsequent implementation of the system could not proceed. In addition, payments to the vendor were done only when the audit team had certified that they had met all the requirements for that particular milestone. ASS#10 noted that "our audit is very strict, every night they will check each work station and inspect the work progress and condition". ASS#11 said, "We have an independent audit team ... you [vendor] complete how much work, the audit team must validate first before we pay you [vendor]".

The key members of the IT project and the vendors were mandated to stay onsite during the entire system development period. The project was expected to run around the clock. ASS#12 said, "We are stationed at the worksite 24 x 7 and if we see any problems [during the development] we will communicate with them [vendors] directly. To facilitate communication, they rented a house that is near the worksite".

The leaders of the IT project team had to lead by example to demonstrate the genuine commitment needed for the project to complete on time. Frequent site inspections and weekly meetings attended by the leaders across all levels were also conducted to instill a sense of urgency and cultivate the mindset of getting things done right on the first attempt. These progress meetings were usually held on weekends, and site inspections were often conducted at night. Often, these leaders had to work longer hours than their subordinates.

... some leaders will follow us to inspect the work site [at night]. This means that there is no rest day, it is like 24 x 7 ... if you have to deal with some situations at home, you can go back to settle it fast ... but basically at night you will be back to the work site (ASS#13).

As a result, the IT project team was able to keep the schedule on track.

4.5. Airport Data Center System

The implementation of the airport data center system started late 2007 and was delivered by March 2008. The system functioned to collate all data related to the terminal's operations from the various systems so as to provide information to aid the decision making process, and to provide a single authoritative source of bill calculation for all the internal stakeholders of Beijing Capital International Airport (BCIA). The system also generated the billing details that had to be error-free, or the organisation would lose the trust of the main users (i.e., the airlines, the shop owners in the terminals, etc).

This project was proposed because there are a lot of data sources and outlets that are controlled by the various departments [in BCIA] ... as a result, during BCIA management meetings, conflicting data was being reported ... we need to have a commonly agreed upon financial data that all stakeholders agree upon ... otherwise you can't bill [these stakeholders] (ADCS#1).

This system played multiple roles, such as supporting the revenue-generating activities for BCIA and subsuming some of the key functionalities currently executed in the existing enterprise resource planning (ERP) system and the terminal operations systems. As a result, some inevitable political conflicts and resistances rose from the staff and the users:

... if we send an inaccurate bill to the airlines, they will say this figure is incorrect. Why? The airlines will say, we wouldn't tell you why, you go and figure it out yourself (ADCS#2).

For an airport of such size, finance is always a headache. The statistics department often came up with figures that didn't tally with the finance department's figures (ADCS#3).

The project had to navigate the many uncertainties that were inherent in managing users' expectations. It also had to tread through the 'political minefields' across BCIA's departments during the system implementation.

When our management gave us this task ... we didn't really know how to work on it because ... we didn't know how to position the system [to appeal to these stakeholders] ... On the other hand, when Terminal 3 is operational, if we don't have this system, revenue collection to our company will be a big problem [to BCIA] (ADCS#4).

4.5.1. Mitigating the Challenges

To address the uncertainties inherent in this project, the IT department appointed a well-respected employee of the airport as the project manager. This project manager was respected for his extensive technical and business knowledge, strong communication skills, good relationship management skills, and improvisation abilities. After assuming this role, he immediately realized the importance of managing the stakeholders' expectations. He executed a comprehensive communication plan designed to align all users' interests toward fulfilling the accounting regulatory requirements mandated by the Chinese Government. Through close co-ordination and many intense negotiations with all the users, the project manager was able to secure a strong buy-in and commitment from all that were involved.

The most important element in this system's implementation is coordinating and managing all the stakeholders of this project (ADCS#5).

When the system goes live, it is inevitable that business processes will need to be changed. Many of these changes, they [users] will feel uncomfortable. But we work together to jointly establish the clarity in the rules of engagement ... and each other's responsibilities throughout the entire process ... when the numbers [financial data] don't tally, we can work together to resolve it amicably (ADCS#6).

Because changes to the system requirements were expected, the project manager instituted a rigorous change management procedure to control the change.

During the Olympics, we do not allow any changes, that is to say all change requests can only be processed during the system testing phase ... for instance if you want to add a new rule in computing your finance billing, then that will require you to submit a change request (ADCS#7).

Having full awareness of the ambitious deadline, the project manager leveraged his good working relationship with the local partners of the outsourced vendor to start the project even before the contract was signed. This had serious consequences because the outsourced vendor, Company CX (a large multinational company), had decided to exit the project (because the company had decided to pull out from the China market) before the project's implementation commenced.

Just before we signed the contract, they [company CX] announced that they don't want to sign the contract because they have decided to pull out from China market ... we took on a lot of risk in implementing the system [before they signed it] (ADCS#8).

The project manager managed to convince the management of the BCIA to award the contract to a local partner of the outsourced vendor, Company LP. He trusted the knowledge and capability of Company LP's project manager, and authorized him complete freedom to implement many of the system requirements. ADCS#9 said, "if we have another vendor, the pressure on us would be really great and we will probably have to work overtime every day".

As a result, this entrusted project manager was constantly willing to do extra work in return. For instance, he was proactive in reporting the status of the project and was flexible in taking up ad hoc work of a smaller scale not specified in the initial terms of contract at no extra cost to BCIA.

They [vendor] are very familiar with the data center knowledge which helps them significantly in communicating with all airlines [users] ... I just tell them the brief overview [of a requirement] and they [vendors] ask a few key questions ... and things get done without us worrying (ADCS#10).

During the system go-live period, one of the engineers [vendor] just had a child. He couldn't sleep well at home. He was always thinking about how to get this system working smoothly until he got a little mentally overwhelmed (ADCS#11).

In contrast, this trusting work relationship was not extended to the users of the system because the project manager did not believe that these users (who were more interested in maximizing their own profit than developing an integrated bill payment system) would serve the best interest of the project. ADCS#12 said, "If you let the users control the process, you will face the challenge of frequently changing requirements...because they represent their own interest".

Weekly progress meetings were held involving all the key users and the team from Company LP. Changes and progress of the project requirement were meticulously tracked and reported during these regular meetings. Payments to Company LP were made only when all the project requirements were endorsed (by the involved users) and fulfilled.

We have meetings. For this project, we have one every week ... every day, the project manager [vendor] will analyze the work progress of every item in this project. Since there is WBS [work breakdown structure] ... you can verify the progress of all the tasks using it [WBS]. Every week they [vendor] need to present the report [during the meeting] (ADCS#13).

Despite the project manager's best efforts to control the changes, users raised three unforeseen change requests related to the billing functionality of the system late into the project. If the requests for change were approved, the project would not have met its deadline. Leveraging his expertise and close working relationships with the users, the project manager managed to persuade the users to postpone the implementation of these requests to the second phase of the project after the Olympics. These measures undertaken kept the project scope in check and thus ensured that the project was delivered on schedule.

5. Discussion

We contend that two contingent factors play crucial roles in developing agile IS development practices for large-scale IT projects; namely, (1) project uncertainty, and (2) project completion urgency. When an IT project is large, uncertainty is an especially salient factor that often troubles even the most experienced project manager (Meyer et al., 2002). This is largely due to many managers' regarding uncertainty as a single type. Thus, they fail to realize that different types of project uncertainty require different management approaches in planning, monitoring, and controlling (Meyer et al., 2002). Very often, these managers assume that most of a project's uncertainties can be identified right from the onset of the project and can be planned for (Meredith & Mantel, 1995). Hence, we analyze our data by first matching the profiles of the four project uncertainties that have been identified in the literature to the large-scale IT projects of our study (see Table 6).

In order for an IT project team to devise an effective IT project management approach to handle project uncertainty, it needs to develop a set of unique capabilities (Fink & Neumann, 2007; Sambamurthy, Anandhi, & Varun, 2003). Hence, we focus on identifying the dominant IT project team capabilities developed specially to address the type of "project uncertainty" identified in each of the four large-scale IT projects in our study.

The other contingent factor crucial to the development of the agile IS development practices is project completion urgency. We postulate that an IT project team needs to develop a set of effective

organizational control mechanisms to manage vendors and users in order to adhere to the required project timeline. For instance, organizational control mechanisms are important factors that affect the software team's ability to respond to changing requirements (Maruping et al., 2009). Hence, our case analysis focuses on identifying the dominant organizational control mechanisms used in each of the four large-scale IT projects.

Table 6. Project Uncertainty Classification for Each IT Project in Our Case Data

IT project name	Project uncertainty type (Meyer et al., 2002)	Definitions of project uncertainty type (adapted from Meyer et al., 2002)	Justifications from case data analysis	Representative supporting quotations
Airport departure system	Variation	Many small influences that will cause the cost, time, and performance levels of a project to vary randomly, but in a predictable range.	Although business process of passenger check-in has remained similar for decades, the main requirement of the system is to perform small customizations tailored for airlines, which vary from one airline to another. This makes the project outcome vary randomly but within a predictable range.	ADS#1
Airport operation database system	Foreseen Uncertainty	A few known factors that will influence the project's outcome significantly, but in unpredictable ways.	The integration between AODB systems in T1, T2, & T3, and the integration between AODB systems and airport transport control system are all identified as major factors that will influence the project outcome. However, the team was unable to ascertain the extent of these influences.	AODB#1 AODB#2
Airport security system	Unforeseen Uncertainty	One or more major factors that will influence the project's outcome significantly but are not known in advance.	The novelty in developing a system to fulfill the highest security requirements ever in the company's history, the unprecedented scale of implementation and the complex coordination efforts required are three major influencing factors of the project outcome that can't be accurately forecasted in advance.	ASS#1 ASS#2
Airport data center system	Chaos	Unforeseen events that happen during project implementation and completely invalidate the project's target, planning and approach.	The sudden withdrawal of the main vendor, company CX, in the initial phase of the project, and the last minute change requests submitted near the end of the project were all unforeseen events that can completely invalidate the project's target, planning, and approach.	ADCS#3 ADCS#4 ADCS#8

5.1. Developing Precision-Based Agile IS Development Practices in the Airport Departure System Project

In addressing the “variation” project uncertainty type in an IT project, our case data reveals that BCIA’s IT department leveraged two dominant IT project team capabilities; namely, (1) the IS planning capability, and (2) the behavioral conditioning capability. The IS planning capability is defined as the capability to anticipate future change and growth in order to choose the right platform for accommodating the change (Feeny & Willcocks, 1998) and to effectively manage the resulting technology change and growth (Mata et al., 1995). This capability was developed by the IT project team, which comprised the project steering committee, the subject-domain experts from various airlines, and the IT department of BCIA. The behavioral conditioning capability is defined as the capability of an organization to influence its stakeholders in order to exercise appropriate behavior during the task execution (Fink & Neumann, 2007). This capability is enacted in our case data in two ways: (1) the technical leader was able to align all the interests of the members of the project steering committee towards a shared goal of enhancing the passenger departure experience, and (2) the training program for the vendor was conducted to infuse BCIA’s culture, practices, and norms. This is consistent with the extant literature whereby the planning and behavioral conditioning capabilities of an IT project team have been identified as the significant factors that influence its ability to accommodate change in the way the users access and use the information resources (Fink & Neumann, 2007).

The trustworthy work relationship between BCIA’s IT project team and its users was established with the help of the IT project steering committee. This good relationship facilitated task-related outcomes and behaviors during the system or the change request implementation, and allowed it to be accurately defined. In order to foster a stronger relationship between the IT project team and the vendor, applying behavior control mechanisms to limit the vendor’s opportunistic behavior was essential. This observation is consistent with the extant literature on organizational control, which posits that it is preferable for a IT project team to exercise formal control mechanisms during the system implementation when the trust level and the task uncertainty are low (Rustagi et al., 2008). By managing the level of trust in the work relationship, the IT project team imposed two dominant control mechanisms (viz., (1) the formal outcome control, and (2) the formal behavior control) to enhance its ability to plan and exert influence on users and vendors. As these formal organizational control mechanisms become more effective, the IS planning and the behavioral conditioning capabilities of the IT project team were enhanced. This allowed the project task and its associated vendor’s behavior to be more accurately predicted, and thus enhanced the effectiveness of the formal organizational control mechanisms that had been applied (Kirsch, 1997). We can rely on the IT project team capabilities and the organizational control mechanisms that were applied to the vendor and the users to accurately identify the system requirements. Thus, we assert that a precision-based agile IS development practice is developed over time. We define the precision-based agile IS development practices as practices that allow an organization to effectively and accurately sense variation (a form of project uncertainty) and fulfill the requirements of addressing the project completion urgency risk in a large-scale IT project. Figure 4 shows our inductively derived model.

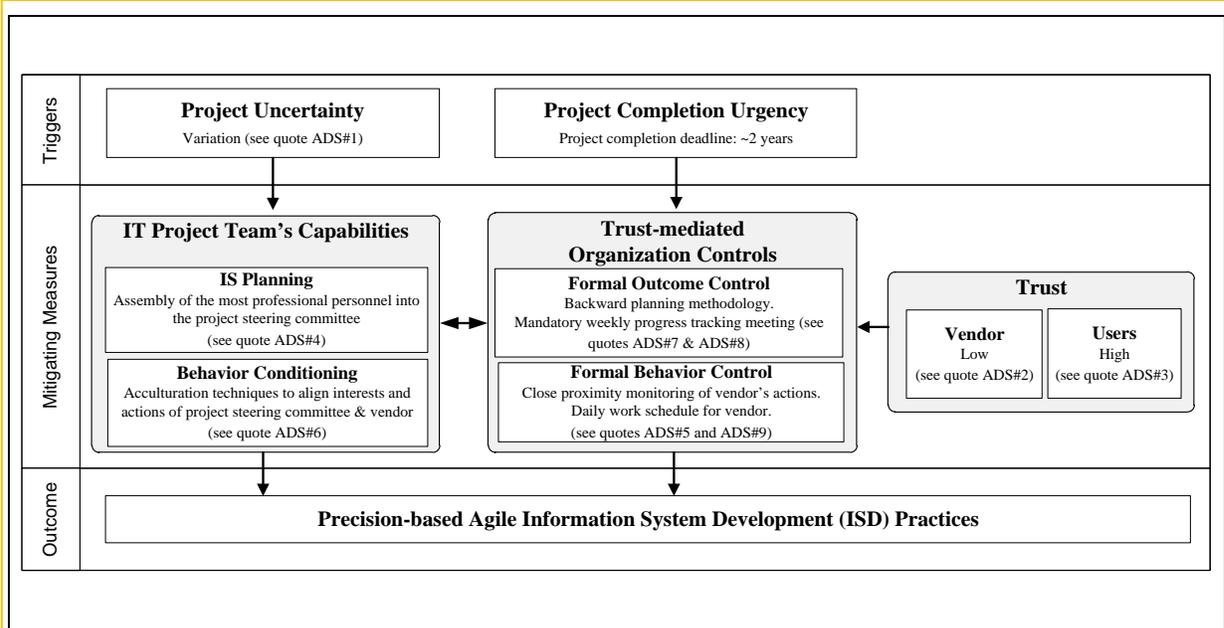


Figure 4. Airport Departure System—Precision-Based Agile IS Development Practices

5.2. Developing Preemptive-Based Agile IS Development Practices in the Airport Operation Database System Project

In addressing the “foreseen uncertainty” project uncertainty type, our case findings reveal that the IT department leveraged two dominant IT project team capabilities; namely, (1) capability reconfiguration, and (2) absorptive capacity. The capability reconfiguration is defined as the capability of an organization to reconfigure its existing capabilities through substitution, evolution, and transformation in response to changes in the environment (Lavie, 2006). Absorptive capacity refers to a firm’s ability to identify, assimilate, and exploit knowledge from the environment (Cohen & Levinthal, 1990). These are capabilities that are best exemplified through consolidating AODB’s experiences of setting up the terminal command centers (capability reconfiguration), assimilating knowledge acquired through constant experimentation, and learning to deliver world-class AODB administration processes (absorptive capacity). The capability of an IT project team to constantly provide high-quality infrastructural services (through capability reconfiguration) and develop in-depth technical knowledge about a system (through absorptive capacity) are identified as significant factors in the team’s ability to be agile during system implementations (Fink & Neumann, 2007).

Our case study also reveals a high level of trust between the IT project team and the AODB administrators (users) and between the IT project team and the vendor. As a result, the two organizational control mechanisms (viz., (1) the informal clan control, and (2) the informal self-control) were applied to ensure the project was completed by the deadline. The informal clan control was most exemplified in the development of a proactive problem solving attitude among the IT project team, the users, and the vendor. The informal self-control was most exemplified in two ways: (1) the AODB administrators were given “freedom” to experiment with the system, and (2) the informal mentor-apprentice arrangements. The literature supports the use of these organization controls in this project for two reasons. First, a controller would typically apply the informal clan control on a controllee when the task-related behaviors and outcomes required during the IS development cannot be clearly specified (Kirsch, 1996). Second, a controller would apply informal self-control on a controllee if the controllee is intrinsically motivated and has engaged in self-monitoring and self-evaluation processes (Kirsch, 1996).

These organizational control mechanisms that are applied to the controllees also helped to strengthen the IT project team capabilities namely, “capability reconfiguration” and “absorptive

capacity". For instance, these informal control mechanisms encouraged the IT project team to experiment and accumulate new knowledge, which, in turn enhanced the IT project team's "absorptive capacity". This helped the IT project team to become more sensitive to changes.

These strengthened capabilities also caused the IT project team to take pride in their work. The team's sense of pride enhanced the effectiveness of the informal organizational control mechanisms that were applied to the team. With the IT project team's capabilities to develop strong problem sensing abilities and the effectiveness of the organizational control mechanisms in nurturing proactive learning and problem solving, we theorize that preemptive agile IS development practices are formed over time. These practices can help to facilitate an organization's effectiveness to preempt the foreseen uncertainty and promote proactive problem solving to address the risk arising from completing a large project on time. Figure 5 shows our inductively derived model.

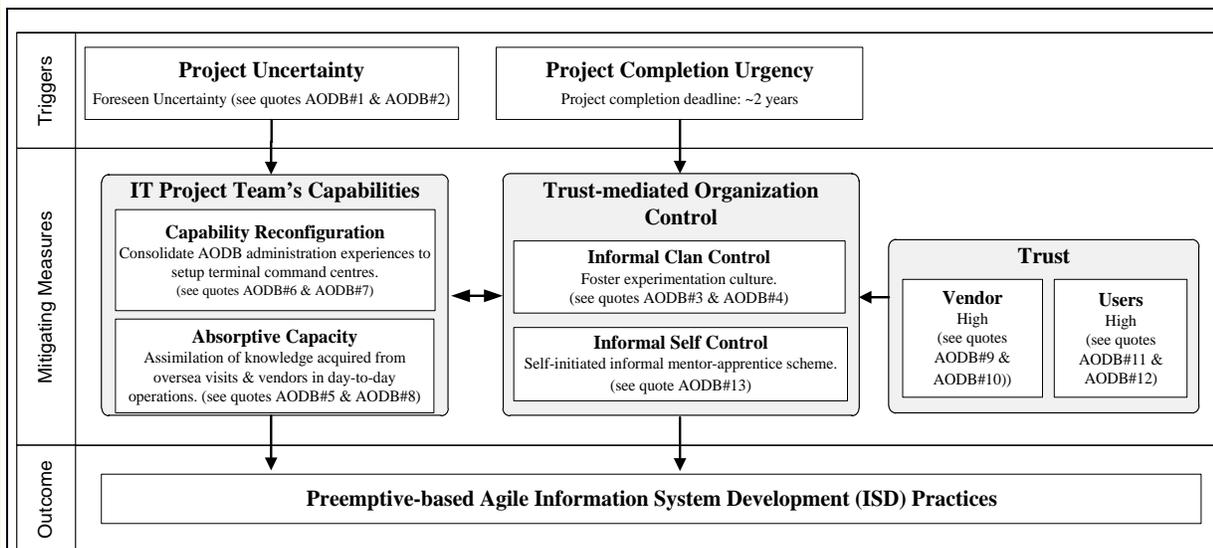


Figure 5. Airport Operation Database System—Precision-Based Agile IS Development Practices

5.3. Developing Adeptness-Based Agile IS Development Practices in the Airport Security System Project

Our case findings reveal that the IT department leveraged two dominant IT project team capabilities (viz., (1) shared ideology, and (2) knowledge integration) to address the unforeseen uncertainty project uncertainty type. Shared ideology refers to the capability to create an attractive identity and a collective interpretation of the reality shared by all the project stakeholders (van Den Bosch et al., 1999). In order to nurture a shared ideology among the project's stakeholders, BCIA's IT project team developed comprehensive communication plans. Knowledge integration is the capability of an IT project team to negotiate, achieve, and refine a shared understanding among all project stakeholders through interaction, sense-marking, and collective learning (Ayas & Zeniuk, 2001; Boland & Tenkasi, 1995). This capability was evident in the IT project team, especially in the initial tender proposal stage, at which it spent a significant amount of time making numerous overseas visits before coming up with the eventual tender specification.

The development of shared ideology and knowledge integration allowed the IT project team to align the expectations of all the project stakeholders towards a common purpose. This alignment of expectations helped to facilitate the easy integration of knowledge and ensure a positive working relationship was maintained among the project stakeholders to deal with the inherent uncertainties throughout the project. Indeed, the literature suggests that the ability of an IT project team to

understand the business environment (through knowledge integration) has a positive effect on the team’s ability to respond efficiently and effectively to changes (Fink & Neumann, 2007). Additionally, the ability to create a collaborative environment (through shared ideology) has a positive effect on a team’s ability to accommodate changes in systems without incurring significant penalties in cost and time (Fink & Neumann, 2007).

When the project implementation commenced, it was evident that the trust level among the IT project team, the users, and the vendor was low. But the IT project team saw the necessity to instill a risk-taking culture in them for a better chance to deliver before the project on time. Two control mechanisms (viz., (1) formal behavior control, and (2) informal clan control) were implemented. These two control mechanisms were best exemplified by the “joint-responsibility deposit scheme” (formal behavior control) and the 24/7 work culture” (informal clan control). We surmise that these controls helped to create an environment that promoted a willingness to take risk and to enact learning behaviors, which, in turn, attributed to delivering the project on time. Indeed, the organizational control literature notes that behavior control is often applied to a situation where the desired behavior during project development can be clearly identified (Kirsch, 2004), and informal clan control is applied when developing a cross-departmental culture for a common purpose is critical to the IS project’s success (Kohli & Kettinger, 2004).

The developed IT project team’s capabilities (namely, knowledge integration and shared ideology), allow the formal behavior and the informal clan control mechanisms to be more effectively applied during project implementation. This is because, when more knowledge was integrated, the IT project team was able to derive more specific and appropriate behaviors. With a shared ideology instilled, the effectiveness of implementing the informal clan control was also enhanced. As these control mechanisms’ effectiveness was enhanced, they improved the IT project team’s capabilities to integrate more knowledge and deepen their shared ideology. The interplay between the IT project team’s capabilities and the control mechanisms contributed over time to the development of the adeptness-based agile IS development practices. In general, these practices allow organizations to develop both an adept capability that encourages calculated risk-taking, learning behavior and effective resource management to address the unforeseen uncertainty, and to cut the risks of a large-scale IT project. Figure 6 shows our inductively derived model.

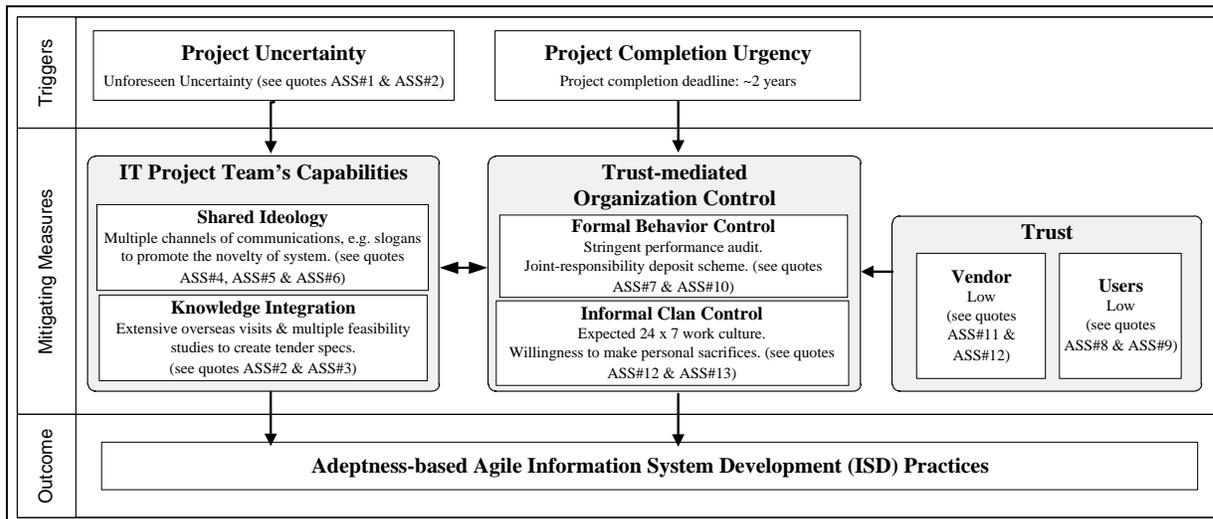


Figure 6. Airport Security System—Adeptness-Based Agile IS Development Practices

5.4. Developing Improvement-Based Agile IS Development Practices in the Airport Security System Project

Last but not least, our discussion on project uncertainty is on “*chaos*”. The IT department was aware that the project would face quite a few human-related issues, such as both the ERP and terminal operations system’s users’ resistance that could potentially jeopardize any part of the target, plan, and approach of the project. As such, the IT department identified two IT project team capabilities (viz., (1) stakeholder management, and (2) IT change management) as the intervention tools essential for the project’s success. The stakeholder management capability is defined as a firm’s ability to manage the linkage between the IS function and the stakeholders outside of the firm, and its ability to align and integrate the processes of the IS function with other functional areas of the firm (Wade & Hulland, 2004). The IT change management capability is defined as the ability of an IS manager to understand how technology can and should be used, and how to motivate and manage IS personnel through the change process (Bharadwaj, 2000). BCIA’s IT project team most vividly demonstrated these capabilities when it instituted the well-respected expert as its project manager and implemented a stringent change management process. These two capabilities are the critical antecedents for an IT project team “to generate more competitive actions and greater action repertoire complexity” (Sambamurthy et al., 2003, p. 244) to cope with the highest level of project uncertainty of the “*chaos*” type.

The effects of “*chaos*” were highly salient during the project implementation. One such effect was the withdrawal of the main vendor, Company CX, from China. Even though the level of trust between the IT project team and the vendor was high, it was expectedly low with its users (due to the potentially opportunistic behaviors of the users when monetary transactions were involved). This difference in the levels of trust induced the use of two control mechanisms (viz., (1) formal outcome control, and (2) informal self-control). While the “formal outcome control” was tailored to ensure a rigorous change management process and align interests of all the users and the vendor toward delivering the system on time, the informal self-control was specially applied to the vendor. The two control mechanisms were most exemplified in the areas of the mandate of the “weekly change” and the progress meetings (formal outcome control), and the “freedom” given to Company LP’s project manager to implement many of the requirements in the project (informal self-control). The application of the outcome controls is also consistent with the extant organizational control literature. We advocate this based on two reasons: for one, the level of trust between the IT project team and its users (i.e., shop owners and airlines) was low; and, for another, the task-related outcomes and impact of every change request submitted by the users could be accurately assessed by both the experienced IT and vendor project managers. The extant literature advocates that, under these two circumstances, an IT project team would likely impose one of the two formal control mechanisms (behavior and outcome control) on the controllee (Rustagi et al., 2008). From our data collection, we note that the vendor’s manager exhibited strong self-monitoring and self-evaluating behaviors. Under such a context, the extant literature supports our observation on the application of the informal self-control organizational mechanism in this project (e.g., Kirsch, 1996).

The application of the formal outcome control helped to enhance the IT project team’s capabilities to manage its large number of stakeholders, whereas the application of the informal self-control helped to enhance the capability of the IT project team to manage change. In return, these enhanced capabilities increased the effectiveness of the organizational control mechanisms applied during the project implementation. The interplay between the IT project team’s capabilities and the organizational control mechanisms contributed to the development of the improved agile IS development practices. These practices allow an organization to develop strong coordination and business-IT alignment capabilities to effectively manage the stakeholders and changes to overcome chaos to mitigate the project completion urgency risk in a large-scale IT project. Figure 7 shows our inductively derived model.

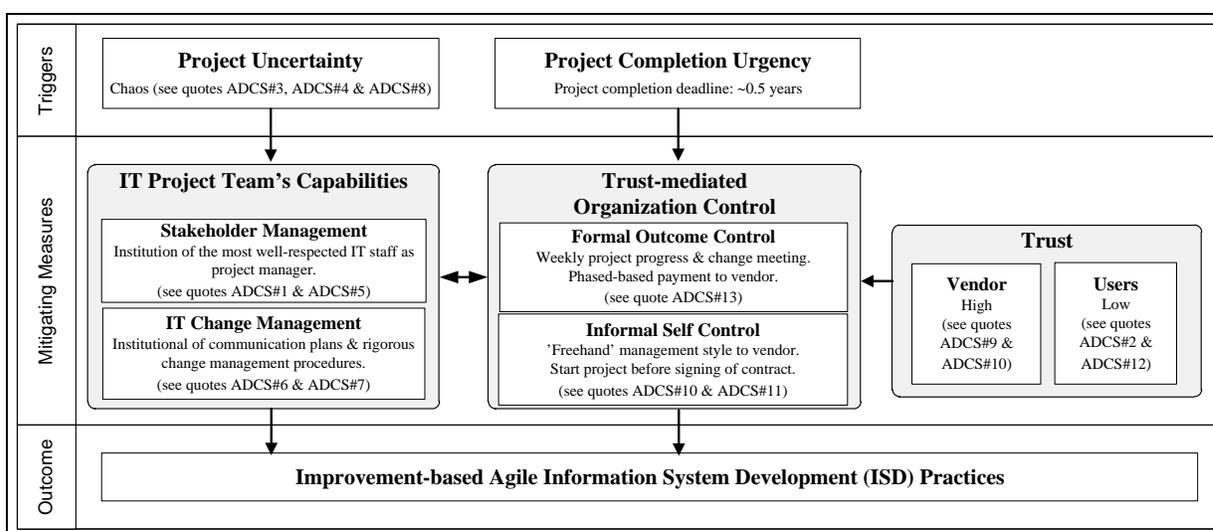


Figure 7. Airport Data Center System—Improvement-Based Agile IS Development Practices

5.5. Developing Sensing-Based & Responding-Based Agile IS Development Practices in a Large-Scale IT Project

Based on our case analysis, we classified the four practices into two types (see Table 7). Our approach of classification is consistent with the extant literature on agile IS development practices as 'agility' in IS development is often defined as having two components: the sensing and responding to changes (e.g., Lee & Xia, 2010). We suggest that, when an organization is subjected to various types of project uncertainty and strict schedules, it can choose to develop one of these practices or a combination of these practices depending on the nature of the uncertainty inherent in the large-scale IT project.

Table 7. Types of Agile IS Development Practices

	Sensing-based agile IS development practices		Responding-based agile IS development practices	
Agile IS development type	Precision based	Preemptive based	Adeptness based	Improvement based
Project uncertainty	Variation	Foreseen uncertainty	Unforeseen uncertainty	Chaos
Definition	A set of agile IS development practices that are sensing centric, in which the focus is to develop capabilities to accurately sense the project uncertainties stated above in order to meet the urgency of project completion and to develop capabilities that allow for proactive response to these uncertainties.		A set of agile IS development practices that are responding centric, in which the focus is to develop capabilities that can flexibly respond to unpredictable project uncertainties stated above in order to meet the urgency to complete the project and to develop capabilities that allow learning from that experience.	

We also checked whether the agile IS development practices identified in our study are indeed agile through the use of Conboy's (2009) agile IS development taxonomy (see Table 8). Our validation result shows that all the agile IS development practices that are inductively driven in our study are considered as agile under Conboy's (2009) taxonomy. This validation is an important undertaking because we do not want to fall into issues such as "practitioner-driven" agile IS development practices, where the definition of "agility" is inconsistent.

Table 8. Conforming to the Taxonomy of Agile IS Development Practices

Taxonomy of agile IS development practices	Trust-mediated organizational control & IT project team's capabilities
Practice components must contribute to one or more of the following: (a) Creation of change (b) Proaction in advance of change (c) Reaction to change (d) Learning from change	We have shown in our models of agile IS development practices that the trust-mediated organizational control and IT project team capabilities are fundamental to (a) the creation of change, (b) the proaction in advance of change, (c) reacting to changes, and (d) learning from them.
Practice components must contribute to one or more of the following and must not detract from any: (a) Perceived economy; (b) Perceived quality; and (c) Perceived simplicity	We have also shown in our models of agile IS development practices that the trust-mediated organizational control and IT project team capabilities increase (a) the overall perceived quality and (b) simplicity in managing IS development, and this, in turn, contributes to the perceived economy (c) of using them for IS development.
Practice components must be continually ready	These identified trust-mediated organizational control and IT project team capabilities have been readily applied to various large-scale IT projects in Terminal 3 including those that are not featured in our study at various degree.

6. Conclusion

By presenting models detailing the four distinct types of agile IS development practices that can be developed in a large-scale IT project, this study makes several important theoretical and practical contributions. First, this study sheds light on the ways that the underlying process of agile IS development practices can be developed in large-scale IT projects. This answers the call by many researchers and practitioners to extend agile IS development research onto large-scale IT projects (Ågerfalk et al., 2009; Dingsøyr et al., 2012; Freudenberg & Sharp, 2010). Second, we have developed our models built on a strong theoretical basis found in the literature. We have built our models from the insight that has been drawn from the research on trust-mediated organizational control, BCIA's IT project team's capabilities, and agile IS development. To the best of our knowledge, we also conclude that our models are the first theoretically driven models on the study of agile IS development practices in the context of large-scale IT projects. Our theoretically driven models help to complement the existing practitioner-driven agile methods, such as XP or Scrum. We are confident that our models will increase the success rate of extending the agile IS development to large-scale IT project in organizations. These contributions are of significant value to both the practitioners and researchers. While prior studies have advocated a contingency view that defines the success of an agile IS development practice in the small-scale projects (e.g., Cao et al., 2009), it is unclear what these contingency factors are for the agile IS development in a large-scale IT project. Our study has also identified two of such contingency factors (viz., (1) project uncertainty, and (2) project completion urgency) drawn from the insight cited in the extant literature. Through our theoretical models, we present ways that the varying types of project uncertainty and project completion urgency can be effectively addressed through the interplay of an IT project team's capabilities and trust-mediated organizational control mechanisms. Finally, in so doing, we provide a trustworthy roadmap for practitioners who are keen to explore agile IS development practices in large-scale IT projects. In addition, our theoretical models provide the necessary foundation for future research on agile IS development practices, especially in the area of large-scale IT projects.

6.1. Future Research Direction & Limitations

As an exploratory study, this study develops theoretical insight to enrich current understanding of agile IS development in a large-scale IT project (an under-explored phenomenon) (Eisenhardt &

Graebner, 2007). Our theories emphasize the distillation of the important theoretical constructs collected in our data that can be used to guide future research and practical implementations of agile IS development practices in large-scale IT projects. A suggestion of a future research direction would be for the theoretical constructs developed in our study to be used in other research methods (e.g. in a survey to achieve statistical generalization). In this regard, we highlight that the theories developed in our study are limited to agile IS development practices in large-scale IT projects that involve the engagement of an outsourced vendor. In the case where a large-scale IT project is done completely in-house, our theories may not be applicable. However, we are certain that our findings would help to enrich the discourse on how to achieve success in the management of large-scale IT projects that are developed completely in-house.

To address the issue that some of the supporting evidence for the theories developed in our study has been obtained from a few individuals and so may hamper its validity, we made significant efforts to develop two interview questioning techniques, similar to those advocated by Schultze and Avital (2011), and an unique interviewing strategy (see Figure 1) that allowed us to generate the richest and thickest data possible. These interview questioning techniques adopted in our study have also been used in other IS studies on IS development (e.g., Avital, 2003). We also followed up with constant validation of the data collected through “member checking”. As such, we can confidently conclude that, by employing these measures, our study is soundly valid.

Acknowledgement

This paper was supported in part by National Natural Science Foundation of China under Grant 70971130 and 71273265. We gratefully thank the department editor and three anonymous reviewers for their insightful and constructive comments throughout the review process.

References

- Abrahamsson, P., Conboy, K., & Wang, X. (2009). "Lots done, more to do": The current state of agile systems development research. *European Journal of Information Systems*, 18(4), 281-284.
- Ågerfalk, P. J., Fitzgerald, B., & Slaughter, S. A. (2009). Flexible and distributed information systems development: State of the art and research challenges. *Information Systems Research*, 20(3), 317-328.
- Ashurst, C., Doherty, N. F., & Peppard, J. (2008). Improving the impact of IT development projects: the benefits realization capability model. *European Journal of Information Systems*, 17, 352-370.
- Austin, R., & Devin, L. (2003). *Artful making: what managers need to know about artists work*. Boston: Financial Times, Prentice Hall.
- Avital, M. (2003). *Reexamining information systems success through the information technology professionals perspective*. Paper presented at the Proceedings of the 11th European Conference on Information Systems, Turku, Finland.
- Ayas, K., & Zeniuk, N. (2001). Project-based learning: building communities of reflective practitioners. *Management Learning*, 32, 61-76.
- Bassellier, G., & Benbasat, I. (2004). Business competence of Information Technology professionals: Conceptual development and influence on IT-business partnerships. *MIS Quarterly*, 28(4), 673-694.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. (2001). Principles behind the Agile Manifesto. Retrieved Dec 30, 2011, from <http://www.agilemanifesto.org/principles.html>
- Beijing Capital International Airport. (2009). *The 52nd gold medal*. Beijing Capital International Airport Co. Ltd.
- Bharadwaj, A. S. (2000). A resource-based perspective on information technology capability and firm performance: An empirical investigation. *MIS Quarterly*, 24(1), 169-196.
- Bhatt, G. D., & Grover, V. (2005). Types of Information Technology capabilities and their role in competitive advantage: An empirical study. *Journal of Management Information Systems*, 22(2), 253-277.
- Boehm, B. (2002). Get ready for agile methods, with care. *IEEE Computer*, 35(1), 64-69.
- Boland, R. J., & Tenkasi, R. V. (1995). Perspective making and perspective taking in communities of knowing. *Organization Science*, 6, 350-372.
- Cao, L., Kannan, M., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems*, 18(4), 332-378.
- Ceschi, M., Sillit, A., Giancarlo, S., & De Panfil, S. (2005). Project management in plan-based and agile companies. *IEEE Software*, 22(3), 21-27.
- Cohen, D., Lindvall, M., & Costa, P. (2004). An introduction to agile methods. In M. V. Zelkowitz (Ed.), *Advances in Computer: Advances in Software Engineering* (Vol. 62). Amsterdam: Elsevier.
- Cohen, W. M., & Levinthal, D. A. (1990). Absorptive capacity: A new perspective on learning and innovation. *Administrative Science Quarterly*, 35(1), 128-152.
- Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, 20(3), 329-354.
- Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11).
- Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.
- Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50(9-10), 833-859. doi: 10.1016/j.infsof.2008.01.006
- Dybå, T., & Dingsøyr, T. (2009). What do we know about agile software development? *IEEE Software*, 26(5), 6-9.
- Eisenhardt, K. M. (1985). Control: Organizational and economic approaches. *Management Science*, 31(2), 134-149.
- Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review*, 14(4), 532-550.

- Eisenhardt, K. M., & Graebner, M. E. (2007). Theory building from cases: opportunities and challenges. *Academy of Management Journal*, 50(1), 25-32.
- Elshamy, A., & Elssamadisy, A. (2006). Divide after you conquer: An agile software development practice for large projects. In P. Abrahamsson, M. Marchesi & G. Succi (Eds.), *Extreme programming and agile processes in software engineering* (Vol. 4044, pp. 164-168). Oulu, Finland: Springer Berlin / Heidelberg.
- Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management*, 16(4), 88-100.
- Feeny, D. F., & Willcocks, L. P. (1998). Core IS capabilities for exploiting Information Technology. *Sloan Management Review*, 39(3), 9-21.
- Fink, L., & Neumann, S. (2007). Gaining agility through IT personnel capabilities: The mediating role of IT infrastructure capabilities. *Journal of the Association for Information Systems*, 8(8), 440-462.
- Finley, I., Wilson, N., & van Huizen, G. (2012). Hype cycle for application development, 2012: Gartner.
- Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile method to software practices at Intel Shannon. *European Journal of Information Systems*, 15, 200-213.
- Freudenberg, S., & Sharp, H. (2010). The top 10 burning research questions from practitioners. *IEEE Software*, 27(5), 8-9.
- Geneca. (2011). Doomed from the start? Why a majority of business and IT teams anticipate their software development projects will fail. Retrieved April 10, 2012, from <http://www.geneca.com/survey/download.php?file=GenecaSurveyReport.pdf>
- Grenning, J. (2001). Launching extreme programming at process-intensive company. *IEEE Software*, 18, 27-33.
- Harris, M. L., Collins, R. W., & Hevner, A. R. (2009). Control of flexible software development under uncertainty. *Information Systems Research*, 20(3), 400-419.
- Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *IEEE Computer*, 34(9).
- Hsu, J. S., Liang, T. P., Wu, S. P. J., Klein, G., & Jiang, J. J. (2011). Promoting the integration of users and developers to achieve a collective mind through the screening of information system projects. *International Journal of Project Management*, 29, 514-524.
- Inkpen, A. C., & Currall, S. C. (2004). The coevolution of trust, control, and learning in joint ventures. *Organization Science*, 15(5), 586.
- Jacobson, L., & Spence, I. (2009). Why we need a theory for software engineering. Retrieved April 16, 2012, from <http://www.drdoobs.com/architecture-and-design/why-we-need-a-theory-for-software-engine/220300840>
- Jiang, J. J., Klein, G., & Chen, H.-G. (2006). The effects of user partnering and user non-support on project performance. *Journal of the Association for Information Systems*, 7(2), 68-90.
- Kettinger, W. J., & Lee, C. C. (2002). Understanding the IS-User divide in IT innovation. *Communications of the ACM*, 45(2), 79-84.
- Kirsch, L. J. (1996). The management of complex tasks in organizations: Controlling the systems development process. *Organization Science*, 7(1), 1-21.
- Kirsch, L. J. (1997). Portfolios of control modes and IS project management. *Information Systems Research*, 8(3), 215.
- Kirsch, L. J. (2004). Deploying common systems globally: The dynamics of control. *Information Systems Research*, 15(4), 374.
- Kirsch, L. J., Ko, D.-G., & Haney, M. H. (2010). Investigating the antecedents of team-based clan control: Adding social capital as a predictor. *Organization Science*, 21(2), 469-489.
- Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, 23(1), 67-93.
- Kohli, R., & Kettinger, W. J. (2004). Informing the clan: Controlling physicians' costs and outcomes. *MIS Quarterly*, 28(3), 363-394.
- Lavie, D. (2006). Capability reconfiguration: An analysis of incumbent responses to technological change. *Academy of Management Review*, 31(1), 153-174.
- Lee, C. H., Venkatraman, N., Tanriverdi, H., & Iyer, B. (2010). Complementarity-based hypercompetition in the software industry: theory and empirical test, 1990-2002. *Strategic Management Journal*, 31, 1431-1456.

- Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS Quarterly*, 34(1), 87-114.
- Mangalaraj, G., Mahapatra, R., & Nerur, S. (2009). Acceptance of software process innovations - the case of extreme programming. *European Journal of Information Systems*, 18(4), 344-354.
- Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, 20(3), 377-399.
- Mata, F. J., Fuerst, W. L., & Barney, J. B. (1995). Information Technology and sustained competitive advantage: A resource-based analysis. *MIS Quarterly*, 19(4), 487-505.
- Meredith, J. R., & Mantel, S. J. (1995). *Project management - A managerial approach*. New York: John Wiley & Sons.
- Meyer, A. D., Loch, C. H., & Pich, M. T. (2002). Managing project uncertainty: From variation to chaos. *Sloan Management Review*, 43(2), 60-67.
- Murru, O., Deias, R., & Mugheddu, G. (2003). Assessing XP at European internal company. *IEEE Software*, 20, 37-43.
- Nambisan, S., Agarwal, R., & Tanniru, M. (1999). Organization mechanisms for enhancing user innovation in Information Technology. *MIS Quarterly*, 23(3), 365-395.
- Nerur, S., & Balijepally, V. (2007). Theoretical reflection on agile development methodologies. *Communications of the ACM*, 50(3), 79-83.
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, 48(5), 73-78.
- Ouchi, G. W. (1980). Markets, bureaucracies and clans. *Administrative Science Quarterly*, 25, 129-141.
- Pacheco-de-Almeida, G. (2010). Erosion, time compression, and self-displacement of leaders in hypercompetitive environments. *Strategic Management Journal*, 31, 1498-1526.
- Pan, S. L., & Tan, B. C. C. (2011). Demystifying case research: A structured-pragmatic-situational (SPS) approach to conducting case studies. *Information & Organization*, 21(3), 161-176.
- Pentland, B. T. (1999). Building process theory with narrative: From description to explanation. *Academy of Management Review*, 24(4), 711-724.
- Rasmusson, J. (2003). Introducing XP into greenfield projects: lessons learned. *IEEE Software*, 20, 21-28.
- Rockart, J. F., Earl, M. J., & Ross, J. W. (1996). Eight imperatives for the new IT organization. *Sloan Management Review*, 38(1), 43-55.
- Ross, J. W., Beath, C. M., & Goodhue, D. L. (1996). Developing long-term competitiveness through IT assets. *Sloan Management Review*, 38(1), 31-42.
- Rustagi, S., King, W. R., & Kirsch, L. J. (2008). Predictors of formal control usage in IT outsourcing partnerships. *Information Systems Research*, 19(2), 126-143.
- Sambamurthy, V., Anandhi, B., & Varun, G. (2003). Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms. *MIS Quarterly*, 27(2), 237.
- Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (2001). Identifying software project risks: An international Delphi study. *Journal of Management Information Systems*, 17(4), 5-36.
- Schultze, U., & Avital, M. (2011). Designing interviews to generate rich data for information systems research. *Information & Organization*, 21(1), 1-16.
- Stephens, M., & Rosenberg, D. (2003). *Extreme programming refactored*. Boston: Addison-Wesley Longman Publishing Co.
- Tarafdar, M., & Gordon, S. R. (2007). Understanding the influence of information systems competencies on process innovation: A resource-based view. *Journal of Strategic Information Systems*, 16(4), 353-392.
- The Standish Group International. (2009). CHAOS Summary 2009 Report: The Standish Group International, Incorporated.
- van Den Bosch, F. A. J., Volberda, H. W., & De Boer, M. (1999). Coevolution of firm absorptive capacity and knowledge environment: Organizational forms and combinative capabilities. *Organization Science*, 10(5), 551-568.
- Wade, M., & Hulland, J. (2004). Review: The resource-based view and Information Systems research: review, extension and suggestions for future research. *MIS Quarterly*, 28(1), 107-142.

- Walsham, G. (1995). The emergence of interpretivism in IS research. *Information Systems Research*, 6(4), 376-394.
- Walsham, G. (2006). Doing interpretive research. *European Journal of Information Systems*, 15(3), 320-330.
- Wang, E. T. G., Ju, P.-H., Jiang, J. J., & Klein, G. (2008). The effects of change control and management review on software flexibility and project performance. *Information & Management*, 45(7), 438-443.
- Wang, E. T. G., Shih, S.-P., Jiang, J. J., & Klein, G. (2006). The relative influence of management control and user-IS personnel interaction on project performance. *Information and Software Technology*, 48(3), 214-220.
- Wang, L. (2007). Agility counts in developing small-size software. *IEEE Potentials*, 26(6), 16-23.
- Weill, P., Subramani, M., & Broadbent, M. (2002). Building IT infrastructure of strategic agility. *Sloan Management Review*, 44(1), 57-65.
- Whetten, D. (1989). What constitutes a theoretical contribution. *Academy of Management Review*, 14(4), 490-495.

About the Authors

Jenson Chong-Leng GOH is a Lecturer in the School of Business at SIM University, Singapore. He received his Ph.D. (Information Systems) from the National University of Singapore in 2012. His research strives to understand how IT can help organizations and society address contemporary challenges, such as information systems management, sustainable development, and business & service excellence, in an increasingly unpredictable world. He adopts an interdisciplinary research approach and has published widely in Information Systems and Computer Science conferences and journals such as *International Conference on Information Systems*; *ACM Special Interest Group on Management of Data*; *Journal of AIS*, and *The International Journal on Very Large Data Bases*. He has over 16 years of experience in managing and strategizing IT in an education industry. He is active in the Singapore's service excellence scene and has won over 20 International and National Service Quality Excellence Awards. He is a certified IT project manager, Lean Six Sigma Black Belt, and TOGAF. He currently serves as a Business Excellence Assessor for Spring Singapore and a National Service Commanding Officer in the Singapore Police Force.

Shan-Ling PAN (Shan L Pan) is an Associate Professor and the Deputy Head of the Department of Information Systems, School of Computing at the National University of Singapore. He is also a visiting professor at the Australian School of Business, University of New South Wales, Australia. His research is centred on the IT and eCommerce industry of China, and the implications of Chinese IT management practices on our understanding of business models and innovation. In the last 6 years, he has conducted over 20 in-depth studies of state-owned enterprises, public organizations and e-commerce start-ups in China. Some of these enterprises include Alibaba.com, China Mobile, COFCO, Haier, Tencent and NeuSoft. He has published more than 80 papers in journals such as *Information Systems Research*, *MISQ Executive*, *Journal of AIS*, *European Journal of Information Systems*, *Information Systems Journal*, *Journal of Strategic Information Systems*, *Information and Organization*, and the *European Journal of Operations Research*. He is also currently an Associate Editor of *MIS Quarterly*, *Information Systems Research*, *Information and Management*, *Communications of the AIS*, and *Electronic Commerce Research and Application*.

Meiyun ZUO is a Professor and Associate Dean in the School of Information, Renmin University of China. He is also the Associate Chair of Chinese Information Economics Society (CIES) and Associate Secretary-General of China Association for Information Systems (CNAIS). He received his Ph.D. from School of Management, Harbin Institute of Technology, China. His work has been published *IEEE Transactions on Engineering Management* and the proceedings of International Conference on Information Systems (ICIS). His research interests include information systems adoption, knowledge management, IT impact on the elderly, and so on.