Association for Information Systems
AIS Electronic Library (AISeL)

PACIS 2008 Proceedings

Pacific Asia Conference on Information Systems (PACIS)

July 2008

INFORMATION SYSTEMS DEVELOPMENT IN A LOW MATURITY ENVIRONMENT: AN EXPLORATORY CASE STUDY ON CONTROL MODES

Ji-Ye Mao
Renmin University of China, jiyemao@yahoo.com

Xia Zhang
Renmin University of China, zhangxia@ruc.edu.cn

Wenmo Song
Renmin University of China, songwenmo@ruc.edu.cn

INFORMATION SYSTEMS DEVELOPMENT IN A LOW MATURITY ENVIRONMENT: AN EXPLORATORY CASE STUDY ON CONTROL MODES

Ji-Ye Mao, Renmin University of China, Beijing, 100872, People's Republic of China, jymao@ruc.edu.cn, jiyemao@yahoo.com

Xia Zhang, Renmin University of China, Beijing, 100872, People's Republic of China, zhangxia@ruc.edu.cn, zxruc@yahoo.co

Wenmo Song, Renmin University of China, Beijing, 100872, People's Republic of China, songwenmo@ruc.edu.cn

Abstract[1]

This paper presents an exploratory case study to examine information systems development (ISD) processes in a low maturity environment, and to understand the role of control mechanisms in project success. The case involves the development of a large scale bank information system, which progressed without a fully-defined "master plan" or much reliance on formal development methodologies, but was successfully launched nevertheless despite some delay. Data were analyzed from the lens of control modes. Results show that clan control emerged as a dominant form of informal control in a high complexity and low methodological maturity environment, and end-to-end user participation through collocation with the developers served as effective outcome control, which appeared to be a critical success factor. The reliance on behavior control was marginal, although the project manager's effective leadership as a form of self-control also played a role in project success. This work contributes to ISD research in general and the development of a control perspective to user participation in ISD. One of the practical implications is that potential problems associated with the absence of formal methods in a low ISD maturity environment may be overcome by strong user control over the process and quality via extensive participation.

Keywords: Information systems development, Control modes, Control mechanisms, Project management, Low maturity environment

[1] Acknowledgments: The authors are grateful to the two anonymous reviewers for their helpful comments and suggestions. This research was supported by a research grant from the national Natural Science Foundation of China (Project No. 70671103).

1 INTRODUCTION

Despite a long history of research on information systems development (ISD), deficiencies still exist in our knowledge about the management of complex development processes (Sambamurthy & Kirsch 2000). In more resent years, it has been suggested that ISD process maturity improves project quality, cost, and timeliness (Adler et al. 2005, Harter et al. 2000). For example, it was found that formal project management practices were more probable for meeting the project schedule, whereas neither project complexity nor project size mattered. In particular, for projects with high complexity, the adoption of formal project management practices was more likely to lead to on-time systems completion (Gowan & Mathieu 2005). The recent popularity of the Capability Maturity Model (CMM) (Herbsleb et al. 1997, Jiang et al. 2004, Ramasubbu 2008) also implies more "mature" forms of software development are desirable.

Not only is control entrenched in formal methodologies for ISD, its critical role in effective project management has also been noted by both researchers and practitioners (Henderson & Lee 1992, Kirsch 1996, 1997, 2002). Control refers to actions taken to regulate or adjust the behavior of the controllee (Kirsch 1996). A series of control modes were introduced along with their antecedents conditions for application in ISD (Kirsch 1996, 1997, 2002). Exercise of control is necessary to effectively manage relationships among stakeholders of multiple organizational units, and to ensure progress by fusing complementary roles and capabilities. However, the primacy of control in ISD has long been questioned with suggestions that methodologies have not been effective or extensively used at all (e.g., Ciborra 2002, Dobing & Parsons 2006). There are calls for rethinking of ISD methodologies, and some even criticized the overdose of sophisticated ISD methodologies and planning, and advocated greater emphasis on care, hospitality and cultivation (Ciborra 2002).

However, much of the prior research on control is conducted in the West, where the maturity of ISD is relatively high. In fact, despite the increasing attention to IS management and implementation in developing countries, few studies have examined ISD processes in this context (Okunoye 2003). In emerging economies, the maturity of ISD is relatively low. For example, a case study on managing outsourced project in a developing country showed, there was only a known project starting date in an outsourcing contract but no specified end date, and without specification of detailed system functions other than some high-level objectives (Okunoye 2003). It is, therefore, logical to expect that in the context of low ISD maturity the control mechanisms might be different than in an environment with high ISD maturity. For example, the adoption of formal project management practices might be particularly difficult for ISD in China, because in the Chinese management culture there is little emphasis on the use of formal co-ordination mechanisms and control structure (Martinsons & Hempel 1995), which is likely the case in many other emerging economies.

This research investigates the following research questions: What control mechanisms are used in a low IS maturity environment that is typical in emerging economies? How do they help the project team overcome difficulties associated with low ISD maturity? This research intends to contribute to the ISD literature in general and to enhance understanding of ISD in low IS maturity environments.

To this end, we conducted an exploratory case study to understand the control modes employed in ISD processes and how they worked. The case involved the development of a large scale bank information system, which progressed without a fully-defined "master plan" or much reliance on formal development methodologies, but was successfully launched nevertheless despite some delay. It is interesting to examine the dominant forms of control and their impact on the project success.

The remainder of this paper is organized as follows. After a review of prior literature on control as the conceptual basis for this research in Section 2, the research method will be reported in Section 3. Then, this paper will describe the background of the case and characteristics of the low ISD process maturity in Section 4. Next, Section 5 presents various control modes that were used to ensure project success, followed by the case analysis. Lastly, Section 6 summarizes the findings, and concludes this paper with a brief discussion.

2 CONCEPTUAL FOUNDATION: CONTROL MECHANISMS IN ISD

The concept of control has occupied an important position in organizational theories over a long period of time. It is generally recognized that effective control mechanisms are needed to help organizations achieve their goals. Kirsch (1996) initially synthesized various control modes and adopted these modes of control to ISD research. This research draws mostly from Kirsch's seminal work on control in ISD (1996, 1997), as a lens for analyzing the case. According to Kirsch (1996, 1997), there are two broad types of control in ISD. First, as a performance evaluation strategy, *formal control* can take two modes, behavior and outcome. Behavior control relies upon specified rules and procedures to be followed for desired outcomes. It could be a detailed systems development methodology, which articulates precise steps for successfully developing a system. The effect of such behavior control depends on how individuals faithfully follow the methodology. Such control is recommended when appropriate behaviors are known and controllees' behaviors are observable. In contrast, outcome control involves rewarding controllees for their meeting of the desired outcome or articulated goals. In the context of ISD, it would mean setting precise target task completion dates and interim milestones. Such control is deemed suitable when outcomes are measurable.

Kirsch also identified two modes of *informal control*, namely clan control and self-control. Clan control "is implemented by promulgating common values, beliefs, and philosophy with in a clan, which is defined as a group of individuals who are dependent on one another and who share a set of common goals" (1997, p. 217). Acceptable behaviors are reinforced through socialization process, rituals and ceremonies. Clan control may be adopted when neither desired behaviors are well articulated nor outcomes are measurable. Moreover, specific goals are typically unknown at the outset of an activity.

Whereas clan control is based on shared norms and values in a team, self-control stems from individual objectives, personal standards and intrinsic motivation (Kirsch 1997). Organizational and individual antecedents of self-control include task complexity, ambiguity in performance evaluation, lack of rules and procedures, and desire to exercise self-control.

It is important to note that both clan control and self-control are independent of other forms of control. For each of the four modes of control, a list of control mechanisms were identified from case studies conducted in the context of traditional ISD methodologies (Kirsch 1997).

Examples of formal control mechanisms include walk-throughs, ISD technical documentation, progress report, project plan, and system testing. It appears formal control mechanisms are usually pre-defined by ISD methodology. It is not clear when traditional ISD methodologies are not used, what control mechanisms would play a dominant role to ensure project success. Furthermore, it would be interesting to explore in a low ISD maturity environment, if there are any new control mechanisms and how they work.

3 RESEARCH METHOD

All of the three authors participated in the face-to-face interviews with the project team, which occurred over a period of about six months in the later stages of the project. During the data collection period, the research team made multiple visits to the development team's offices to conduct in-depth interviews. The interviews were taped recorded, and the researchers also took a large volume of notes. Supplementary documents were requested and given to verify and triangulate our findings. Data were collected from a variety of sources including minutes of over 40 internal meetings of the project team, newsletters, and design documents. The data collection methods are summarized in Table 1

Data Sources	Contents or Informants (number of individuals)
Archives	Achieved materials including the project Feasibility Report, project plan, weekly progress reports, project summary reports
Field Observation	Visits to the development team, in their office and work environments, observation and discussion
In-depth interviews	Project Manager (1), leader of the business group (1), key members of the business group (8), key members of the technical groups (6 sub-group leaders), and the leader of the testing group (1)
Follow-up visits	After data processing and initial analysis, follow-up visits to key informants were arranged to confirm the completeness and accuracy of key processes and preliminary conclusions.

Table 1. Overview of Data Collection Methods

In the process of writing up the case, the authors also had numerous contacts with the project manager via email and online chat for clarifications and supplementary information, and got detailed answers. Clarifications were also made through emails, telephone calls, and instant messenger.

Our analysis focused on identifying and classifying evidence that illustrates the low process maturity and control mechanisms that helped overcome the difficulties associated with low maturity. The data analysis overlapped with data collection, as prescribed in the literature (Eisenhardt 1989, Miles & Huberman 1994, Yin 2003). After the data collection, the recorded interviews were transcribed to facilitate analysis. After plowing through the existing data of transcriptions and case note, strategies were generated for collecting new, and often more relevant, data. Content analysis was iterative by collaboration between the first and second authors from the perspective of control modes. Finally, concepts were linked together in a framework that represented the low ISD process maturity, control mechanisms and system success.

4 CASE BACKGROUND: AN ISD PROJECT WITH LOW MATURITY

Alpha Bank is one of the "big four" state-owned commercial banks in China. The systems, code named "AIS," would be a platform for management accounting, and its users would include finance and accounting employees at all levels within the bank. AIS would also be instrumental for implementing several strategic reforms in the bank.

The feasibility report was completed by a rather unusual team consisting of 13 internal employees, who were experienced domain experts, plus just one IT person only as a technical advisor and facilitator. The composition of the project team in the initial stage set the tone for strong user control and influence. The domain experts were summoned from subsidiary branches partly due to the headquarters' lack of manpower and partly because they would represent well the future users. The systems development relied upon the bank's internal business and technical staff as the main force, supplemented by outsourcing to overcome the lack of technological or project management capabilities with such new and large systems. The approved budget was over 50 million RMB (about US$7 million), involving a project team of over 90 members at its peak time. It took two and a half years to complete the system, and despite some delay and much struggle, AIS was rolled out smoothly and generally well received.

In the AIS project, there were clear signs of lack of formal methodology and low maturity in ISD. The project progressed in a chaotic and exploratory manner in the initial stages without a "master plan" or structure of a methodology, as illustrated in the following aspects.

Evolving Project Leadership and Team. When the project team was officially formed, it had 16 IT developers; all of them came from different provincial branches, plus 13 subject domain experts, who were conducted the earlier feasibility study. Three managers formed the initial project management team, which lacked unison and effectiveness. However, several months later two of them were transferred out of the project for promotion and other reasons, leaving behind Mr. Niu, who was able, strongly motivated, and devoted, as the sole project manager.

The project team was overwhelmed by the challenge to use a new platform to develop a huge complex system, and this was coupled with the enlargement of scope and number of subsystems. Three months after the kickoff of the project, over a dozen developers along with a project manager from XYZ Company joined the project team. Later, when the need for more developers arose, senior management at the bank picked two other smaller software companies for the development team as outsourcers, which created significant challenge for managing the project team. The number of developers on the team exceeded 70 in the coding stage.

Development Platform Change. There existed no consensus on the development platform and tools within the team from the beginning. Most of the developers were familiar with the traditional C language based on the UNIX platform and the client/server architecture. The few developers who had participated in e-Banking projects recommended the newer Java language based on the browser/server architecture, but gained no support from the rest of the team.

The team spent its first month on refining requirements and regrouping the subsystems, but was unable to make a decision on the development approach and platform. It was also unable to decide on documentation management. For most people, it was the first time to face such a mega-size system, there was no consensus on the modeling tool for the business logics and transactions, methods to describe the business processes, and necessary templates. The team developed and compiled a series of templates of their own for various documentation needs, e.g., for product design specification, and functional modeling. They decided to use the familiar data flow diagrams (DFDs) to model the complex business processes. The first two months passed away, without much progress.

When the team refined the requirements and regrouped the subsystems, it became clear that the JEEE technology based on the browser/server architecture was a better choice for leveraging the Internet technologies, simplifying systems development, promoting data sharing and reuse, and consolidating system functions. However, due to the mismatch with the team's technical skills and lack of leadership, there was an impasse.

As the understanding of the requirements deepened, the project team fully recognized that the traditional development method, technologies, and architecture could not meet the requirements. Meanwhile, through constant discussion, the domain experts and technical members refined the requirements. As a result, the five previously defined subsystems were restructured into seven, and each of them was given redefined modules and functional specifications. The structural enhancement further revealed the incompatibility between the traditional approach and the advanced functional requirements. The project team finally decided to move onto the JEEE platform.

After the completion of the detailed design, the development groups had another debate on the documentation standard for implementation. This time, under the leadership of Niu, the team converged on adopting the UML standard. The team invited members with extensive UML experience to train the rest of the team. This was followed by design based on UML.

Management of Outsourcing Partners. The involvement of XYZ was expected to inject into the team not only much needed technical experience with the JEEE platform but also project management expertise. However, to Niu's surprise, the XYZ developers seemed to have not worked together before, without a common process and style. Their two group leaders were not able to create a harmonious and collaborative group culture, and the group had shown signs of disintegration, which planted the seed for later labor problems. To further complicate the situation, the two leaders from XYZ even got into a power struggle, competing against each other. Niu decided to assign the latecomer to the project supervision group, leaving the first leader in the platform group. "*Since XYZ developers did not get along with each other well, they were split to different development groups, and this was also thought* *helpful for them to build a collaborative relationship with the domain experts*," the project manager indicated.

Restructuring of the Team Structure. A dilemma faced by the project leaders was the selection of each group leader. A domain expert would not be comfortable with the technologies, whereas the outsourced developers did not know about the business and they might meet resistance from the bank employees. As a compromise, each group had two leaders, a technology leader from the external partner and a business leader from the bank, at the risk of creating overlaps in authorities and uncovered ground, and higher communication costs associated with a larger number of group leaders.

Major aspects of the low maturity exhibited in the AIS project are summarized in Figure 1 below.

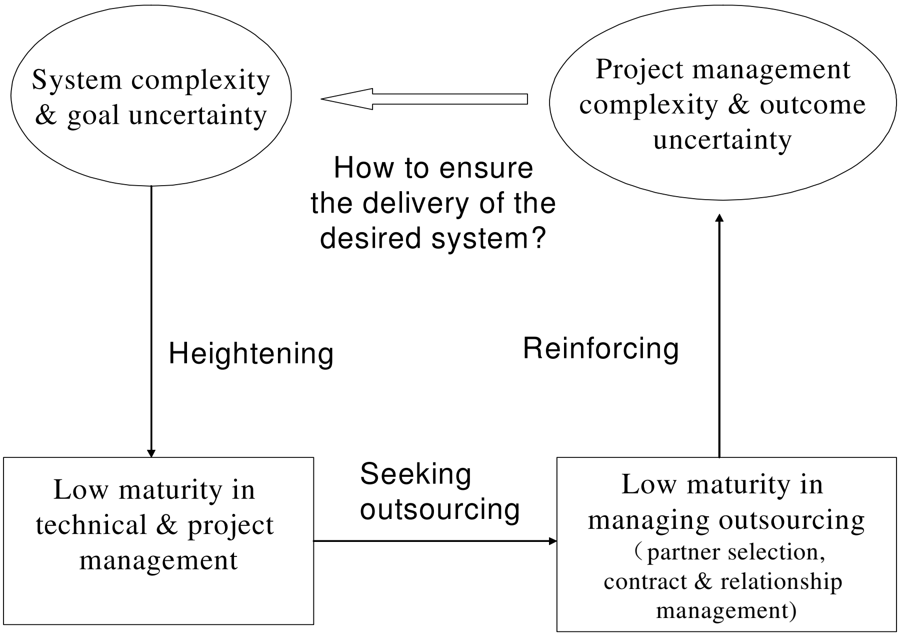
System complexity & goal uncertainty
Project management complexity & outcome uncertainty
How to ensure the delivery of the desired system?
Heightening
Reinforcing
Low maturity in technical & project management
Seeking outsourcing
Low maturity in managing outsourcing (partner selection, contract & relationship management)

Figure 1. Summary of aspects of low maturity and their relationship

5 DATA ANALYSIS: VARIOUS FORMS OF CONTROLS AT WORK

Our analysis of the AIS development process shows that it heavily relied upon informal control, including some identified in the literature and some emergent ones specific to the case, whereas the use of formal control was limited and marginal by comparison. Our analysis of informal control focuses on clan control for the most part.

5.1 Informal Control - Clan Control

Top Management Support. There were clear signs of clan control from the very top in the bank. Senior management in the headquarters signaled its commitment and strong backing to the project. The President of the bank and a Vice President took the leadership roles of the steering committee. The general managers of the bank's technology unit and business unit became members of the committee. They made frequent visits to the development site to give morale support to the development team, and handed out bonuses to the outstanding contributors. Top management support was mainly in the form of resources allocated in particular. Budget control was a low priority, as internal personnel costs were not tracked and no limit was set for travel and some other expenses.

People-centered HR management. Given the diverse sources of the team members, Niu had to deal with a whole series of human resources issues with each group and yet with little resource at his disposal. He found that on the paper, the XYZ developers were JEEE experts, but some members' technical skills were short of expectation and most importantly they could not collaborate as a team. Later, "*another thorny issue surfaced. Developers from XYZ started to slack off in a concerted manner, after several times of internal conflicts and controversies due to the heavy workload and pressure associated with the coding stage*," according to the project manager. At one point, nearly 20 of them went on strike, which posed a major threat to the already delayed schedule. Niu had to step in to be the middleman between the two parties, and helped to put an end to it. To the outsourced developers, Niu's principle was to "s*how respect and blend with them.*" When XYZ developers initially worked in their own group, there was a tendency of communication breakdown with the domain experts. Since the restructure, personnel from the two sides got better communication and improved their attitudes. To Alpha's internal domain experts, Niu had no effective mechanism of influence, as their income and promotion decisions were beyond his scope. To the developers from the regional branches, "*Niu could only rely upon his personal charisma, and use intangible means to motivate the troop*," one of the team members observed.

Learning and Mutual Learning. One of the distinct features of the AIS project was that there was substantial learning of business domain knowledge by the developers. Before the systems design stage, developers were given training on the business model. Each module must designate dedicated technical and business personnel to this process. The business domain experts responded to questions raised by the developers, and this was done frequently and conveniently due to the collocation of technical and business personnel. The project team held daily meetings to discuss the progress and arising issues for each module. The technical leader and business leader of each module were required to attend such meetings, which were seen as an opportunity for the two sides to learn from each other. It was considered important for them to learn about modules being developed by other groups.

Communication. Because of cultural differences, Alpha employees and outside developers had different styles of work, and they tended to blame the other side when problems arose. Niu fully realized that "*communication alone would not be enough to solve the problems, but ambience of trust was key to generating central gravity and team cohesiveness. In fact, to the outside developers, financial reward was less important than respect.*" Therefore, he advocated treating them as Alpha employees. They organized numerous social functions such as hiking on the weekend and group-based competitions, and treated the outsourced developers equally to give them a sense of belonging.

5.2 Self-Control by the Project Leadership

In the absence of methodological maturity and strong process management, the project success relied upon the extraordinary effort by one or more individuals, to a large extent. In the AIS project, many of the organizational and structural problems were dealt with by a committed and competent project manager, who exercised strong self-control. His sense of responsibility and tireless work made a difference. His leadership style was primarily relationship oriented (Ehrhart & Klien 2001), as he emphasized social contract within the project team, treated members with compassion and empathy, showed respect and appreciation.

5.3 Formal Control

Behavior Controls - Incremental Job Specification. When the requirements analysis began, the project team did not have a well specified organizational structure or detailed job specifications. Based on a rough division of work, the team was organized into three groups, product development, architecture, and project management group, with some overlap of work. It was only at a rather late stage that the project manager took measures to modify the interim targets and organizational structure dynamically and increasingly specific, and eventually reached a comfortable state through trial-and-error.

Outcome Controls. One of the most distinctive features of the AIS project was that the large number of domain experts, who were summoned by administrative order from many local branches of the bank, were collocated with the developers as a core part of the team and participated in the project from end to end. Other than in the coding stage, when they prepared testing cases, they were full participants as the source of requirements throughout all other stages of development. They contributed to the requirements definition for the system, construction of the system model and architecture, functional point analysis, prototype of user interfaces, all the way from use cases to test cases at both ends of the development process. Equipped with sign-off authorities, they played an important role in insuring the project a success (as shown in Figure 2).

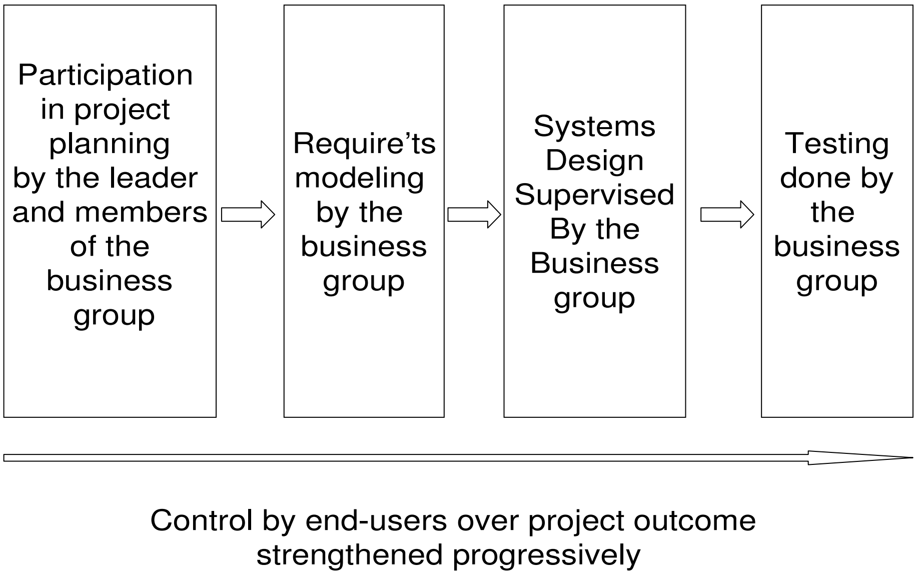
Participation in project planning by the leader and members of the business group
Require'ts modeling by the business group
Systems Design Supervised By the Business group
Testing done by the business group
Control by end-users over project outcome strengthened progressively

Figure 2. Outcome control exercised through user participation

The inadvertent adherence to the user-centered design principles (Barkley & Saylor 2001, Mao et al. 2005) proved to be an extremely useful form of outcome control. Through this mechanisms, design were received frequent feedback and the domain experts and developers could consult each other in regular meetings and informally. The desired outcome by the front-line users was secured in this manner.

When asked about the benefits, a developer commented, "*since there was no similar product in place before, it was difficult to envision the final product. Therefore, close interaction between the developers and the user experts was really important. Luckily, in our setup, the interaction was very good.*" "*Domain experts' feedback started from the prototype, and they had plenty of communication throughout the design process,*" another developer added. Others also noted that *"because they also took the role of testers, they could easily detect any deviation in the systems development from the right course.*" The domain experts were also aware of the important contribution that they made. One of them remarked, "*our participation shaped the evolution of the system design. For example, there was no such a module as project management design. Related functions scattered in many different modules. It was a gradual realization to the team that it would be better to cluster them all together. Consequently, we decided to have an independent module for project management.*"

6 FINDINGS AND CONCLUSIONS

6.1 Findings and Discussion

Given the large scale, huge complexity, and high uncertainty of the project, it was difficult to specify the final outcome at the outset, or to specify a set of ideal behaviors. The low maturity in the ISD processes was exhibited through the evolution of project leadership, switches in development platforms including the programming languages, modeling tools, and documentation methods, and multiple restructuring of system functions and modules. As a result, it would be difficult to enforce formal methodologies in such a low ISD maturity environment. Informal control mechanisms were more likely to be adopted, which is consistent with Kirsch's predication (1996, 1997).

More specifically, three modes of control were found to have contributed to project success in the case. First, in response to the challenges arising from the scale, complexity, and evolving requirements, the project team resorted to informal control, mostly clan control including strong backing from the top management, charismatic leadership of the project manager, people-centered process management, learning and mutual learning between technical and IT personnel, cross module learning, and extensive user participation. Clan control made up the inadequacy in weak behavior control.

Second, one of the most extraordinary features of this case is the end-to-end participation by domain experts as part of the project team. The advantages of such full and effective user participation can be elaborated from the following two perspectives on outcome control. On one hand, the bank's back-office transactions were very complicated, and it was impossible for the technical team to fully understand the comprehensive business logics and workflows in the usual one-shot approach to requirements analysis. Domain experts' feedback was essential for timely correction of any diversion from the overall objectives of the systems. On the other hand, collocation enabled frequent and constant communication between the technical team and business team. As one of the developers commented, "*the costs for collecting requirements are usually very high, but in our case the communication cost is really low*." In a way, as a form of strong outcome control, a series of micro adjustment prevented any radical deviation from the desired system design and implementation.

Third, the project manager showed a high degree of self-control in his effort to design a pragmatic scheme for project management via trial-and-error to a large extent. The human resources practice and relationship maintenance played an important role in the project success. Effort in cultural blending between the internal and external team members was instrumental for clan control.

Lastly, it is worth exploring the prerequisites for the exercise of various control modes. The bank was completely in charge of the project management, including major design decisions and resource allocation, and yet without stringent financial constraints. In other words, the bank took full responsibilities and ownership, which ensured end-to-end user participation in the process. Otherwise, it would not have been possible to maintain on site a large team of domain experts, who willingly conducted many tasks normally done by the technical group, such as preparing the feasibility report, requirements analysis, preparation of test cases, and system testing.

As to the limitations of this research, whereas it is hoped that findings from this research may be generalizeable, the primary limitation of this study is that the single case may not be enough to make general inferences. Moreover, in light of the unique cultural background of this case study, cross-cultural generalizeability is a question mark and subject to verification in other cultural contexts and different degrees of ISD maturity.

6.2 Conclusions

This work contributes to ISD research in general and the understanding of control mechanisms in a low ISD environment. The main contribution of this case study is that it identified various forms of control that helped overcome potential problems associated with low process maturity. Problems arising from project complexity and uncertainty were dealt with strong clan control. The lack of experience and guidance of formal methodology was balanced and compensated by a project team with strong clan control and self-control (exhibited by the project manager in particular). Outcome control was effectively achieved through extensive participation by domain experts collocated with the developers. The domain experts acted as both the source of domain knowledge, and supervisors and quality control to protect user interests. As illustrated in the case, the combination of both strong clan control and outcome control could ensure project success, in contrast to behavior control embedded in formal ISD methodologies.

One of the most interesting findings of this research is the high effectiveness of outcome control achieved through end-to-end participation by domain experts, who served as permanent members of the development team and made sure constant feedback was available to the developers. The practical implication is that potential problems associated with the absence of formal methods in a low ISD maturity environment may be overcome by strong user control over the process and quality via extensive participation.

References

Alder, P.S., McGarry, F.E., Irion-Talbot, W.B. and Binney, D.J. (2005). Enabling Process discipline: Lessons from the journey to CMM level 5. MIS Quarterly Executive, 4(1), 215-227.

Barkley, B.T. and Saylor, J.H. (2001). Customer-Driven Project Management: Building Quality into Project Processes. McGraw-Hill Professional; 2nd edition.

Ciborra, C. (2002). The labyrinths of Information. Oxford University Press.

Dobing, B. and Parsons, J. (2006). How UML is used. Communications of the ACM, 49(5), 109-113.

Ehrhart, M.G. and Klein, K.J. (2001). Predicting followers' preferences for charismatic leadership: The influence of follower values and personality. Leadership Quarterly, 12, 153-179.

Eisenhardt, K. M. "Building Theories from Case Study Research", *Academy of Management Review* (14:4), 1989, pp. 532~550.

Gowan, J.A., Jr, and Mathieu, R.G. (2005). IS project management: project characteristics and management practices. Journal of Enterprise Information Management, 18(1/2).

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

Henderson, J. C. and Lee, S. (1992). Managing I/S design teams: A control theories perspective. Management Science, 38(6), 757–777.

Herbsleb, J., Zubrow, D., Goldenson, D., Hayes, W. and Paulk, M.C. (1997). Software quality and the capability maturity model. Communications of the ACM, 40(6), p30-40.

Jiang, James J., Klein, Gary, Hwang, Hsin-Ginn, Huang, Jack, and Hung, Shin-Yuan (2004). An exploration of the relationship between software development process maturity and project performance. Information & Management, (41:3), 279-288.

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), 215-239.

Kirsch, L. J. Sambamurthy, V. Ko, D. and Purvis. R. L. (2002). Controlling information systems development projects: The view from the client. Management Science, 48(4), 484-498.

Mao, J., Vredenburg, K., Smith, P.W., and Carey, T. (2005). The state of user-centered design practice. Communications of the ACM, 48(3), 105-109.

Martinsons, M.G. and Hempel, P.S. (1995). Chinese management systems: Historical and cross-cultural perspectives. Journal of Management Systems, 7(1), 1-11.

Miles, M. B. and Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook,* Sage Publications, Beverly Hills, CA.

Okunoye, A. (2003). Large-scale sustainable information systems development in a developing country: The making of an Islamic banking package, Annals of Cases on Information Technology, 5, 168-180.

Ramasubbu, N., Sunil, M., Krishnan, M. S., Kemerer, Chris F. (2008). Work dispersion, process-based learning, and offshore software development performance. MIS Quarterly, 32(2), 437-458.

Sambamurthy, V. and Kirsch, L.J. (2000). An integrative framework of the information systems development process. Decision Sciences, 31(2), 391-411.

Yin, R.K. (2003), Case Study Research: Design and Methods, (3rd Ed), Sage Publications: Thousand Oaks, Cal.