ECIS 2009 Proceedings

European Conference on Information Systems (ECIS)

2009

Blogging to express self and social identities, any one?

Vupong Nguyen
The National Australian University, vuong.nguyen@anu.edu.au

Walter Fernandez
The National Australian University, walter.fernandez@anu.edu.au

BRIDGING THE GAPS: PACKAGED SOFTWARE IMPLEMENTATION PROJECTS IN VIETNAM

Vuong Nguyen, The National Australian University, vuong.nguyen@anu.edu.au

Walter Fernández, The National Australian University, walter.fernandez@anu.edu.au

Abstract

Implementations of advanced information systems in developing countries are often perceived as a way to economical prosperity and modernity. However, even in the best of scenarios implementing major information systems is far from a risk-free activity. The reality of developing countries adds extra complexity to an already demanding endeavor. This study explores socio-technical aspects of an unprecedented technological journey in the Vietnamese banking industry, and thus responds to calls for more empirical research in this field. In this paper we present contextual issues affecting the studied software package implementation project and propose a theoretical framework for a basic social process we called "gap bridging."

Keywords: Packaged software, Implementation projects, Developing countries, ICT for Development, Grounded theory, Case study, Empirical studies, Theory building.

1 INTRODUCTION

Businesses in developing countries often adopt advanced information technology (IT) as a way of keeping pace with the developed world. This is both a necessary and a risky activity. While IT investments in developing countries can contribute and foster economic growth and development (Avgerou, 2003, Yoo, 2003), IT implementation projects are intrinsically prone to failure (Lyytinen and Hirscheim, 1987) and these failures are more significant in developing countries (Heeks, 2002).

The socio-technical conditions of developing countries demand special research attention. On one hand, there is an increasing interest in the use and adoption of information systems (IS) in these countries (Walsham and Sahay, 2006). On the other hand, increasing the knowledge of conditions and processes related to socio-technical changes in developing countries through the adoption of technology is significant to both IS research and practice (Avgerou, 2008). Yet, the number of empirical studies on IS implementation in developing countries is limited (Walsham et al., 2007).

In response to this, our study investigates the implementation process of large package-based information systems (Iivari, 1990) in Vietnam. This study presents a theoretical account that emerged from the perspective of Vietnamese client organizations engaged in "an organizational effort to diffuse and appropriate information technology within a user community" (Kwon and Zmud, 1987 p.231).

2 LITERATURE REVIEW

IS implementation is an important research topic that has been studied since the 1970s (Lucas, 1975, 1981, Ginzberg, 1981, Cooper and Zmud, 1990, Lapointe and Rivard, 2007). While “the early implementation research lacked a sufficiently rich technological, institutional, and historical context” it succeeded in identifying critical success factors and providing multiple insights into the development and implementation process (Lucas et al., 2007 p.208). After its initial concerns with success and failure, the direction of research on IS implementation moved to packaged solutions, enterprise-wide implementations and rapid implementation (Lucas et al., 2007).

Packaged IS implementations have been studied since the 1980’s (e.g., Markus, 1983, Lucas et al., 1988). Packaged software presents benefits over custom-made software; however, the implementation of packaged software is not trouble-free (Iivari, 1990) and organizations are likely to find hidden costs (Lynch, 1984), multiple pitfalls (Lynch, 1985, Ben, 2005) and implementation risks in the external business context, the organizational context, the IS context and the project context (Scott and Vessey, 2002).

IS implementation projects in developing countries add risks to four contexts identified by Scott and Vessey (2002), presenting an interesting area of research with many possibilities. IS adoption in developing countries has been mainly studied from the perspectives of technology transfer and diffusion, social embeddedness, and transformative discourses (Avgerou, 2008). However, these studies can also focus on the role of IS in development, cross-cultural collaborations, localization, and marginalization (Walsham and Sahay, 2006).

The change agency aspect of technology must also be considered. The IS implementation process clearly involves organizational change (Zmud and Cox, 1979, Markus, 1983, Orlikowski, 1993, Lyytinen and Newman, 2008). Markus and Robey (1988) suggested that theories about organizational changes caused by technological innovation should be evaluated in terms of causal agency, logical structure and level or unit of analysis.

Following Markus and Robey (1988) causal structure dimensions, our research adopts an emergent perspective, with the logical structure of process theory at organizational level. This exploratory, theory-making, study follows an interpretive epistemology to analyze empirical data from case studies of large packaged IS implementations in Vietnam.

3 RESEARCH METHODOLOGY

Our objective is to generate a substantive theory of packaged IS implementation in Vietnam while allowing the special economic, political and developmental aspects of Vietnam to emerge without the constraints of theoretical preconceptions. We adopted classic grounded theory methodology (Glaser and Strauss, 1967) to derive theory from case study data (Fernández, 2004). Grounded theory (GTM) is an inductive, contextual and procedural method suitable for exploratory studies (Orlikowski, 1993, Myers, 1997) that facilitates the generation of substantive theory based in the discovery of patterns from participants’ accounts and data collected in the field (Glaser, 1998). GTM fits well with case studies (Eisenhardt, 1989, Orlikowski, 1993), provides well established guidelines for data collection and analysis and is likely to produce a theory that is relevant, rigorous and readily modifiable (Glaser, 1978, Fernández and Lehmann, 2005). Using case-based data for theory building studies allowed us to deal with complex processes enacted by multiple actors in a real life situation and to use multiple sources of evidence to derive knowledge (Eisenhardt, 1989).

3.1 Data collection

It was critical to the success our study to have access to IS projects that were important to local organizations and exhibited organization-wide impact. To gain access to suitable sites personal contacts were needed. The first author attended a banking conference in Vietnam; from contacts made at this conference, we gained access to six cases. To facilitate theoretical emergence, we choose to start with the case of *CORE*, a significant IT project at *BetaBank* (pseudonyms) because it was "the most complicated of the real cases" (Yin, 2003 p.74) and thus well suited as a *foundation case* (Fernández, 2004). BetaBank is a top commercial bank in Vietnam and CORE presented an excellent foundation case given its magnitude, socio-technical complexity. CORE was also the first system of its kind to be introduced in Vietnam and it was under implementation from 1994 to 2004.

Following Yin (2003), we collected data from documents, records, interviews, and direct observations of the systems in use and the actors *in situ*. The first author spent 4 months at BetaBank. The first month allowed for familiarization with the case through document reading, observations and informal discussions with project personnel. This activity was very important to become familiar with the case and also for the participants to become familiar with the researcher. The following three months were used for formal interviews; yet, access to documents and informal "chats" continued.

Interviews were the primary source of information. Data from 15 formal interviews were collected and analysed. We interviewed different stakeholders including the bank's board members, IT department managers and IT staff, relevant business department heads and members, project manager and project team and business unit users. Each interview lasted up to two hours, with average of one and a half hours and was recorded and transcribed, producing more than 250 pages of text.

Following the GTM approach, the interview process evolved over time as the themes emerged from data analysis. For example, our earlier interviews were more open and unstructured, but the later ones were guided by the emergent themes and concepts developed from data analysis (Glaser, 1978). Many brief follow-up discussions were also used to gather further information, to explain emerging issues or to clarify aspects or issues found during data analysis. The strategy to develop rapport during the first month was very effective in facilitating the engagement of participants.

3.2 Data analysis

Data analysis activities included open coding, sorting, selective coding and theoretical coding (Glaser and Strauss, 1967, Glaser, 1978). Open coding produced 217 codes related to meaningful excerpts from the data (or *instances*). The codes were further sorted into categories while the constant comparison of incidents facilitated the generation of theoretical memos, thus raising the conceptual level (Glaser, 1978). By comparing incidents in each category, we generated theoretical properties of the category. The emerging 26 themes included concepts related to gaps, communication, learning, conflicts, goals, delays and performance.

We conducted further theoretical sampling and selective coding within the core themes. Memo writing proved to be helpful in recording the ideas generated during the analysis and theorise on interrelations. Memos also help to guide our research discussions and to reach higher levels of conceptualization. From this iterative process the "core variable" (Glaser and Strauss, 1967) emerged, we called it *the gap bridging process*, a basic social process (Glaser, 1978) described in section 5 of this paper.

4 BETABANK'S CONTEXT

This section describes factors in the external and internal environment of the bank that we found influential in the case. The external environment presented the following influential factors:

- *Laws and regulations.* The banking services were heavily regulated by the state laws and regulations. The CORE implementation was affected by requirements that were mostly suitable to the manual operations of old-fashioned commercial banks. The fast pace of integration of Vietnam into regional and world economies demanded changes in laws and regulations. These changes were often sudden and unexpected.

- *Industry development.* In the late 1980s, Vietnam's banking system was transformed into a two-tier system, separating the state bank and commercial banks. Consequently, in the 1990s the banking industry in Vietnam was less developed than those in Asian region and industrialized countries. The CORE project was then seen as the pioneer in modern banking technology and most Vietnamese commercial banks were observing how BetaBank implemented CORE.
- *Technological base.* In the early 1990s, most of the commercial banks used either their in-house developed program or package programs. These programs could only be used in a local area network for each branch with its sub-branches. Data exchange between a bank's headquarter (HQ) and its branches were through file transfer by modem connection over telephone lines. The national telecommunication infrastructure was in poor condition with low capacity and interrupted services.

BetaBank is one of the four biggest local commercial banks in Vietnam with over 5,000 employees and a branch network covering over the country. It provides wholesale and retail banking services to government agencies, corporate customers and the general public. The bank had strong banking services to support foreign trade, FDI, ODA and remittance. The organizational context had a direct effect on the CORE project. The most influential factors in this context included the followings.

- *Business strategies.* Before CORE, each branch operated as an independent bank with its own funds, management and operations. In the early 1990s, the bank's management started a process of centralization of fund management and increased control over the branches' operations; as we shall explain, this was a key driver for the CORE project.
- *Business processes.* Most of the business processes at the bank were poorly-documented. Policies relating to business processes or operations issued by HQ were usually ambiguous, causing different interpretations. Business processes were not standardized across the bank; for example, a loan to a small enterprise would be provided differently across branches. HQ was unable to ensure the observance of its policies by the branches.
- *Management control.* As a consequence of a long tradition of branch autonomy, the organizational structure of each branch developed independently and was different from that of HQ. Each functional department (either at branch or at HQ level) had considerable authority and strong views regarding how their specific business should be conducted. The HQ management control was weak, and rewards and penalties were ineffective and seldom used.
- *Corporate culture.* Following the autonomous traditions, BetaBank's culture was characterized by a focus on local interest and unwillingness to share information and knowledge. The function-based organizational structure contributed to the lack of cooperation and cross-organizational team spirit.
- *IT strategies.* The bank did not have clear IT strategies. The bank had a tradition of in-house IT provision of all IT supports to business areas as required by the business operations.
- *IS/IT management policies and procedures.* Before the project, all the big applications at the bank's network were developed by the IT department at HQ. Each branch had its own IT group to support the branch's operations and to develop applications for its own needs.

5 THE CORE BANKING SYSTEM PROJECT

In the early 1990s, while BetaBank was working with partner banks in the developed world, the bank realized that they needed to improve its operations through the use of IT. As a consequence, the bank's top management decided to implement a *core banking system* (the CORE project). This was the first core banking system (CBS) to be implemented in Vietnam.

The adopted CBS was packaged software developed in USA and used by many commercial banks in USA as well as in some Asian countries. The CORE project went through a sequence of partly overlapping phases (see Figure 1) classified by the critical events and characteristics of the activities undertaken by the organization and project implementation team.

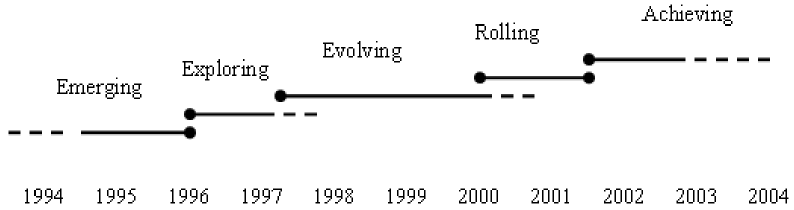

Figure 1: Phases of the CORE project.

In the *emerging* phase, BetaBank decided to use a CBS for online service and central processing of transactions for the whole bank, including HQ and its 25 branches. CORE was perceived by top management as a mechanism both to modernize operations and to bring the branches under control. The IT department initiated the project and assisted the bank's management in making important decisions such as choosing the client-server technology and buying a modern packaged system rather than developing in-house. In early 1996, a vendor from Malaysia was chosen to provide and implement a software package solution developed in USA. The functionality of this system was thought to be aligned with the business objective of centralization of the fund management undertaken by the bank.

At the *exploration* phase, a project taskforce was setup to study the base system provided by the vendor and also the bank's business processes and operations. The taskforce performed a gap analysis and prepared the user requirement document (URD) for customizing the system. The group encountered several barriers, including: their own lack of knowledge and experience in the technology and modern banking practices; lack of cooperation and contributions from other units at the bank; and, lack of business and IT strategies to guide the development of the URD. Hence, their URD was too general and failed to capture the bank's operations in sufficient details. Based on this deficient document, the bank and the vendor signed a contract to customize and implement the CBS across BetaBank. A flexible contract condition covered unanticipated changes required by state regulations.

The *evolving* phase was the longest and most difficult for the bank and the vendor. The formal implementation team was set up, consisting of IT staff and business staff from HQ and some branches. The interactions between technical changes and business changes made the implementation more complicated. The technical side required an iterative and evolving process of system customization/development, system testing and gathering further user requirements. The business side required changes to the business processes, organizational structure, policies and procedures were proposed, discussed, and implemented. Also, the implementation teams from client and vendor, this phase required the involvements of many other groups, including the bank's top management, functional departments, IT department, and users at branches. This phase experienced poor planning and a significant lack of formal and suitable methods. The changes were characterized as an *emergent process*. Conflicts were frequent in most tasks and among different stakeholders involved. Solving conflicts was seen as a time consuming process that caused delays in the project. Fortunately, the learning was most effective for the implementation team and other stakeholders. This phase lasted over three years and ended with the launch of the new system at a single branch at the bank's HQ.

The forth phase, *rolling*, aimed at the installation and use of the systems in all of the bank's branches. This was a hectic process with insufficient time to perform complex tasks such as user training, data conversion and launching of the system. This phase started in early 2000 and lasted over 1 year with 24 branches converted from the in-house developed legacy systems to the new CBS. While the changes to business operations were implemented, the training was ineffective and users were unable to exploit the benefits from the new system causing delays in benefit realisation, as explained next.

The last phase, *achieving*, took place from mid 2001 to the end of 2003. During the first year after the rollout of the system, the bank-wide use of the new system demanded errors corrections, fine-tuning of the system, and changes to business processes. The bank started to achieve benefits from the new system, experiencing growth in customers and services offered. BetaBank was able to introduce many new financial services thanks to the technological foundation of the new CBS.

From the data analysis the initial framework emerged as a pattern that was relevant and core to the activities of the CORE project. The participants main concerns were able to be explained by a *basic social process* (BSP) that we call *Gap-bridging* (for an extensive discussion on BSPs, see Glaser, 1978). The following section presents this emerging BSP as a theoretical framework.

6 THE EMERGENT THEORETICAL FRAMEWORK

From data analysis following grounded theories methodology, we discovered the emergent theoretical framework of package information system implementation in the context of Vietnam – the gap-bridging process (Figure 2.). A discussion on the nature and role of each sub-process is presented next.

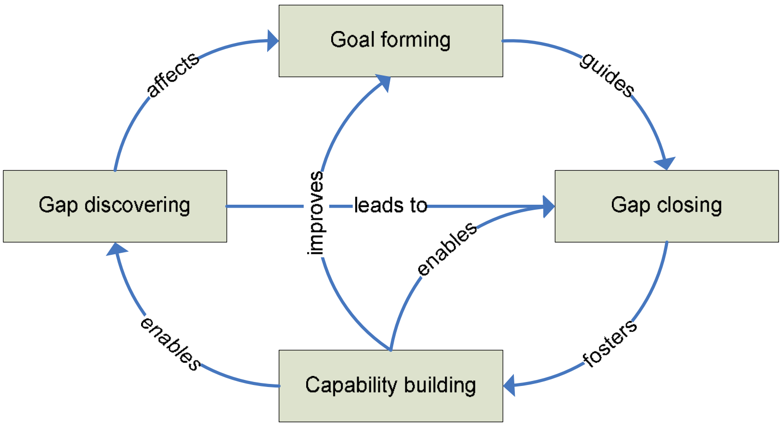

Figure 2: The emerging Gap-bridging basic social process.

6.1 Gap discovering

Gap discovery is a social process that detects discrepancies between the expected reality and the reality found in action. We observed many instances where the implementation teams discovered differences between the system purchased and the bank's needs. One common cause for discrepancies related to contextual differences between USA and Vietnam, as one team member said:

> The development of the software was based on the thinking and logics of the foreign country [USA]; therefore what we needed to do was to customize totally. The technology of had to be changed totally.

The implementation of the CBS started with the gap discovering process, yet, the gap was initially perceived in technical terms. Another team member defined gap analysis as follows:

> Gap analysis is the comparison between the new system and the old [legacy system] to identify what is the same and what is different.

Yet, our data suggests that gap discovering refers to gaps in the intersection of context and system, with social and technical aspects. Thus, the gap discovering process serves to initiate the negotiation process that will enable the resolution of the gap and the progress of the project. This process detects mismatches between the characteristics of the system and the local context, including environmental, organizational and user contexts (Figure 3).

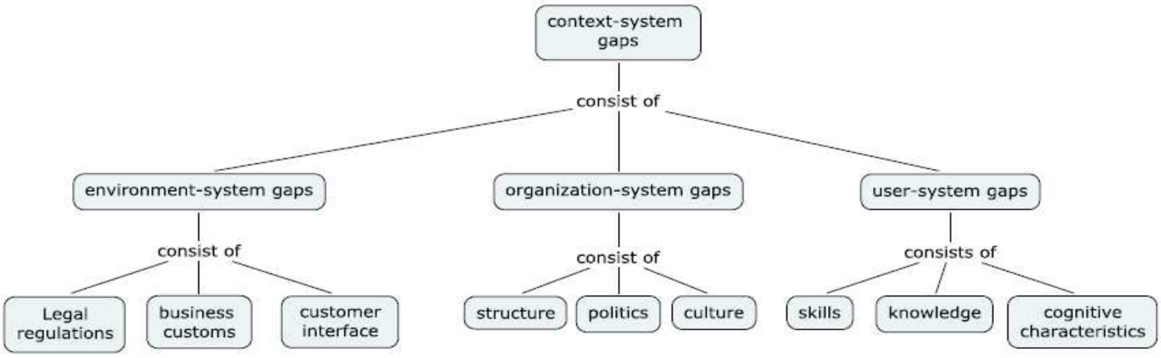

Figure 3: properties of the context-system gaps (CORE data analysis).

Environment-system gaps refer to the mismatches between the system characteristics and the environment of the organization in which it is to be used. We found three elements (or gaps) at this level, including legal regulations, business customs and customer interfaces.

At the organization-system level, the gaps are mismatches between the proposed system and the existing structures, politics and culture. Gaps regarding the organizational structure are often easier to detect, they refer to elements such as communication channels, control mechanisms, decision rules and chain of commands (Markus and Robey, 1983). However, gaps related to organizational politics and culture tend to remain latent (undiscovered) for an extended period. Organizational politics refers to the distribution and use of power to influence decisions (Barbara and Sonny, 1985). Power relations among individuals and groups in BetaBank were important to the project and also difficult to discover and solve. The third element, organizational culture, “a complex set of values, beliefs, assumptions, and symbols that define the way in which a firm conducts its business” (Barney, 1986) acted as a *retardant* of information sharing, acceptance of alien technologies and collaborative behaviours.

Finally, user-system gaps refers to discrepancies regarding system user interfaces and operational procedures that required a certain level of user’s competence, skills, knowledge and certain cognitive characteristics. For example, assumptions about adequacy of training effort were proven incorrect because computing skills and literacy among the bank employees were lower than the vendor’s assumptions. This resulted in frequent user-induced errors, a more protracted implementation cycle, reduced user acceptance, and a more difficult user support task.

The gap discovering process is a cognitive process that converts unknown gaps into known gaps at all three levels. In the CORE project, both BetaBank and the vendor intended to carry out the gap discovering process during the *Exploring* phase in order to develop a complete specification of user requirements. However, this intension failed due to natural limitations in human cognition. Clearly, to imagine the unknown under the influence of the known and with imperfect information is a toll-order, as a team leader attested:

> [t]he development of URD, because of the thinking of the old ways and not grasping of all the complexity of the project, was very limited. We found that it is really funny when we read that URD now.

As seen in this indicator, the gaps were obvious, even funny, when observed with the benefit of the experience. Yet, the actual gap discovering was an on-going, incremental, process that went along with the gap closing process throughout the implementation resulting in a more suitable system. As one team member put it:

> [t]he [system] was customized over the years into something totally different from the initial URDs. … [W]hen we actually worked on the program, the development of URD never stop. For over 10 years, it is still being developed.

The delays in the process of customization were caused by several factor including the following: (a) the *organization-system* gap was very complex and contained tacit knowledge, embedded in the organization’s contexts (at structural political and cultural levels) and in the CBS; (b) the project implementation teams lacked bank-wide support to investigate the organization's contexts; and (c) the project team consisted of young, inexperienced, people with limited knowledge of the bank's existing operations and management. Prior to the CORE project, the implementation team had no exposure to modern CBSor to an IT implementation project; they discovered their way while doing. One team leader stated:

> "We did this project for the first time and had no training or education on how to carry out such a big project in a professional way. Mostly, we learned by doing."

It could be argued that a more comprehensive identification of context-system gaps through the initial implementation team's investigation could have contributed to forming more clear and realistic project goals and to a more efficient gap closing process. Yet, the observed *gap discovering* process was an important process leading to *gap closing* throughout the implementation, and had an important role in influencing the *goal forming* process.

6.2 Goal forming

The *goal forming* process consisted of three interrelated sub-processes: the *identifying (needs and constraints),* the *balancing* and the *evolving* processes, which are next explained in more detail. The discovery of gaps often caused the implementation team and BetaBank's management to struggle with the choice of keeping the bank's existing business processes or followings the functions and processes in the vendor's base system. Consequently, one of the emerging themes was *need-identifying*. This process required the implementation team to clarify expectations and desires of both the bank's management and the system users in order to achieve more precision in identifying clarify the goals. However, the requirements of these two groups were often tacit and divergent, demanding of the implementation team a great deal interactions to discover hidden details.

Another emerging process was the *constraints identifying* process, whereby the implementation team realized that not all identified needs could be satisfied with the available time, resources or because of external factors. Only when the identified needs were balanced with the identified constraints, feasible goals could be formed. However, *balancing* needs and constraints was an imprecise science requiring significant skills and knowledge. During the early stages of the project the goals were vague and not uniformly understood among the project stakeholders. As a result, the project team and management often underestimated the constraints, resulting in unrealistic expectations or unrealisable goals. As the project progressed, the project goals became clearer and more realistic; the efficacy of the goal forming process increased as the team's capability improved overtime.

Implementation goals are important elements, they guide the change process by specifying what, when and how the system or the organization should be changed. The data showed that *evolving* goals acted as a guide to the gap closing process, which is discussed next.

6.3 Gap closing

Gap closing is defined as the process of creating organizational fit (Iivari, 1992) between the system and the organization by eliminating the discovered context-system gaps or by reducing the gaps to an acceptable level. This process consists of two sub-processes: *changing* and *conflict solving*.

Based on our data, we agree with Orlikowski and Hofman (1997) in that "when organizations are using information technologies to attempt unprecedented and complex changes … the depth and complexity of the interactions among theses activities is only fully understood as the changes are implemented." The observed change in the CORE project provided evidence for three types of changes: *emergent*, *anticipated* and *opportunistic*.

At a high level of abstraction, the change process consists of *technological change* and *business change*, with each following its own process and interacting with the other. The change to the packaged software was initially planned to be done by sequential steps, but the actual method used was iterative process of requirements, customizations, testing and further requirements. The cycle continued until the users accepted most of the functions in the system.

The lengthy customization process was influenced by the following factors. First, the users, including the end users and functional departments, were not involved in the earlier phases, but they had most important voice in testing and accepting the system. Only at this phase, the users fed their requirements to the customization along their testing of each system function by test cases. Second, the requirements of the laws and regulations were found to be also a significant source of changes to the customization. Considerable delays were caused by ambiguous legal requirements to banking services and accounting procedures discovered during the customization stage and during tests.

The business changes included organizational structure, business processes, management control and IT management. In this domain, the implementation team regularly interacted with different functional departments and top management. The prominent issues here were related to knowledge gaps, management bureaucracy and lack of responsibility and accountability. Several attributes influenced this process, including the following: (a) differences in understanding between groups about certain implementation issues, (b) the difference between the realities embedded in the organization and the software system and the perceptions of the people, (c) the management bureaucracy resulting in a lengthy decision making process that involved many functional departments and different management levels, (d) multiplicity of implicit procedures for different types of issues made the process more complicated, and (e) unwillingness to accept responsibility causing delays in decisions.

Change was often a source of conflicts among the groups involved in the implementation. Change resistance existed at all levels of the organization and appeared in all phases of the implementation process. Conflict existed between the change promoting (i.e., the implementation team) and change resistant groups (e.g., management, functional departments, users and vendor's team). Changes also brought uncertainties and at times altered the distribution of power and benefits among the groups involved. In pursuing their different self-interests, these groups created conflicts. Lastly, gaps in knowledge caused misunderstandings and opposing views regarding implementation issues.

In the CORE project, conflicts happened in almost all issues emerging from the implementation process. The project implementation team was central in the conflict resolution process, balancing the different goals and interests of top management, users and vendor. Yet, satisfying the parties was difficult, as the following quotations illustrate:

> ..not all disagreements on the issues was solved satisfactorily. … [W]hen there were requirements from BetaBank, they [vendor] asked why we needed in that way. Both two sides were tense with each other.

> … the technology and IT department …tries to keep the functions in the base system as much as possible. The users tried to keep as what they were doing. There were battles between the two. … Even in the board of management, there were fights on these issues.

With unclear responsibilities and accountabilities and a tendency to delay decision, the main conflict resolution method was a compromising process by which the involved parties went through lengthy discussions and negotiations to arrive at short-term solutions. As one informant put it:

> [o]ur opinion was that one thing was necessary, we needed to persuade the other side about that, but you could not force them understand in our way. It was also true that they could not force us to understand in their way. It was human aspect.

The continuous nature of the change process coupled with the constant resolution of conflicts fostered the necessary learning in the project for all parties involved. The numerous issues emerging during the implementation required constant discussion and negotiations among the groups, closing the discovered gaps contributed to building critical capabilities in the implementation team.

6.4 Capability building

The implementation team was central to the implementation process. Their capability consisted of knowledge, experience, and skills that contributed to the deployment of CORE. Team capability at any point in the project is the results from a dynamic process involving staff turnover/recruitment and learning process.

Knowledge, experience and skills were limited at the beginning of the project, affecting both the client and vendor implementation teams. The client's team members were either newly graduates recruited for this project or staff from functional departments with limited working experience in BetaBank. This caused certain level of frustration, as one of the users stated:

> [the client team's] operational knowledge and experience was an issue. When they received our complaints and requests, they could not understand why they should be as we requested.

In addition, the team knowledge of modern banking operations and management and of the capabilities of the CORE technology was also lacking. Fore example, a team member stated that:

> All the people in the project team, when joined the project, did not know about the modern banking operations embedded in the new program. We had to investigate the system to understand the technology as well as those embedded business processes.

On the other hand, the vendor's team was knowledgeable in the technology but had no knowledge, experience and skills in implementing such project in Vietnam. This problem would have affected any vendor as the project was the first implementation of a CBS in Vietnam. Thus, both implementation teams had significantly reduced capabilities at the beginning of the CORE project, the only way out for them was by learning in action.

However, learning during the implementation process was challenging for the individuals and groups involved in the project. They faced work overload, uncertainties, language difference, and staff turnover. To overcome the challenge, the learning process emphasized on learning focus, learning from others and learning by doing, as this quotation indicates:

> We did not have education or training [on IS implementation]; therefore we learnt by doing and self adjusting. ... We exchanged ideas, for example, something we did but we heard that people said it was not reasonable and should be done like this or that. We had to think thoroughly what people said was reasonable or not. We adjusted the way of working for my self and for the whole group in the direction that we believed it would be better.

Learning happened during the change process in both technological domain and business domain and also during conflict resolution of the many issues that the client and the vendor had to confront.

Another key element to build the capability of the implementation team was its recruitment process. As the project developed through the phases, the implementation team changed in size and expertise composition. While recruiting more staff contributed to capability building, staff turnovers decreased the team's capability, as experience and knowledge in the project were lost.

The implementation team was a key actor in the gap closing process. By proposing and implementing changes and solving the conflicts throughout the project, the team not only enabled the delivery of a complex project in difficult circumstances but also became knowledgeable and effective.

7 CONCLUSION

The paper discussed the process of packaged IS implementation in the context of Vietnam and confirmed the importance of the context to the implementation process, indicating areas for attention. This case study contributed to knowledge in the area of IS implementation in developing countries. By using grounded theory with case-based data, the study developed a theoretical framework called "gap bridging" with four sub-processes – gap discovering, goal forming, gap closing and capability building. We used the framework to explain the roles and characteristics of these sub-processes as well as how they were interrelated. It also provides evidence of issues confronted while implementing and using application packages as means of modernisation of business practices.

However, this is only an *emerging* framework developed during the first phase of a major study, thus the framework needs more empirical work to further refine and develop its properties and produce a substantive theory. To achieve theoretical saturation, we are currently working with data from two subsequent cases. From the integration of these data with the relevant extant literature, we will propose a substantive theory on package IS implementation in developing countries.

The research in progress reported in this paper highlights an important set of processes that need to be considered when software packages developed and used in the developed world are implemented in developing countries. The topology of context-system gaps and gap discovering process developed in this study may be useful for client organizations in selecting and implementing such application packages. Moreover, lessons learned from the proposed gap-bridging process can be utilized to overcome prominent issues in countries such as Vietnam, and thus facilitating technological adoption in the developing world.

References

AVGEROU, C. (2003) The Link between ICT and Economic Growth in the Discourse of Development. IN M., K., R., M. & A., P. (Eds.) *Organizational Information Systems in the Context of Globalization.* Boston, Kluwer Academic Publishers.

AVGEROU, C. (2008) Information Systems in Developing Countries: a Critical Research Review. *Journal of Information Technology,* 23**,** 133.

BARBARA, G. & SONNY, S. A. (1985) Politics and Strategic Change across Organizational Life Cycles. *Academy of Management Review,* 10**,** 707.

BARNEY, J. B. (1986) Organizational Culture: Can It Be a Source of Sustained Competitive Advantage? *The Academy of Management Review,* 11**,** 656-665.

BEN, L. (2005) Potential pitfalls in packaged software adoption. *Commun. ACM,* 48**,** 119-121.

COOPER, R. B. & ZMUD, R. W. (1990) Information Technology Implementation Research: A Technological Diffusion Approach. *Management Science,* 36**,** 123-139.

EISENHARDT, K. M. (1989) Building theories from case study research. *Academy of Management Review,* 14**,** 532-550.

FERNÁNDEZ, W. D. (2004) The Grounded Theory Method and Case Study Data in IS Research: Issues and Design. IN HART, D. N. & GREGOR, S. D. (Eds.) *Information Systems Foundations Workshop: Constructing and Criticising.* Canberra, Australia, School of Business and Information Management, Faculty of Economics and Commerce, Australian National University.

FERNÁNDEZ, W. D. & LEHMANN, H. P. (2005) Achieving Rigour and Relevance in Information Systems Studies: Using grounded theory to investigate organizational cases. *The Grounded Theory Review: an international journal,* 5**,** 79-107.

GINZBERG, M. J. (1981) Key Recurrent Issues in the MIS Implementation Process. *MIS Quarterly,* 5**,** 47-59.

GLASER, B. G. (1978) *Theoretical Sensitivity: Advances in the Methodology of Grounded Theory,* Mill Valley, Calif., Sociology Press.

GLASER, B. G. (1998) *Doing Grounded Theory: Issues and Discussions,* Mill Valley, Calif., Sociology Press.

GLASER, B. G. & STRAUSS, A. L. (1967) *The Discovery of Grounded Theory: Strategies for Qualitative Research,* Chicago, Aldine.

HEEKS, R. (2002) Information Systems and Developing Countries: Failure, Success, and Local Improvisations. *The Information Society,* 18**,** 101 - 112.

IIVARI, J. (1990) Implementation of In-house Developed vs Application Package Based Information Systems. *SIGMIS Database,* 21**,** 1-10.

IIVARI, J. (1992) The Organizational Fit of Information Systems. *Information Systems Journal,* 2**,** 3-29.

KWON, T. H. & ZMUD, R. W. (1987) Unifying the fragmented models of information systems implementation. IN R. J. BOLAND, J. & HIRSCHHEIM, R. A. (Eds.) *Critical issues in information systems research.* John Wiley & Sons, Inc.

LAPOINTE, L. & RIVARD, S. (2007) A Triple Take on Information System Implementation. *Organization Science,* 18**,** 89-107.

LUCAS, H. C. (1975) *Why information systems fail,* New York,, Columbia University Press.

LUCAS, H. C. (1981) *Implementation : the key to successful information systems,* New York, Columbia University Press.

LUCAS, H. C., WALTON, E. J. & GINZBERG, M. J. (1988) Implementing Packaged Software. *MIS Quarterly.,* 12**,** 537-549.

LUCAS, H. C. J., SWANSON, E. B. & ZMUD, R. (2007) Implementation, Innovation, and Related Themes Over The Years In Information Systems Research. *Journal of the Association for Information Systems,* 8**,** 5-10.

LYNCH, R. K. (1984) Implementing Packaged Application Software: Hidden Costs and Challenges. *Systems, Objectives, Solutions,* 4**,** 227-234.

LYNCH, R. K. (1985) Nine Pitfalls in Implementing Packaged Applications Software. *Information Systems Management,* 2**,** 88 - 92.

LYYTINEN, K. & HIRSCHEIM, R. (1987) *Information Systems Failures a survey and classification of the empirical literature*, Oxford University Press.

LYYTINEN, K. & NEWMAN, M. (2008) Explaining information systems change: a punctuated socio-technical change model. *European Journal of Information Systems,* 17**,** 589-613.

MARKUS, M. L. (1983) Power, Politics, and MIS Implementation. *Communication of the ACM,* 26**,** 430-444.

MARKUS, M. L. & ROBEY, D. (1983) The Organizational Validity of Management Information Systems. *Human Relations,* 36**,** 203-225.

MARKUS, M. L. & ROBEY, D. (1988) Information Technology and Organizational Change: Causal Structure in Theory and Research. *Management Science,* 34**,** 583-598.

MYERS, M. D. (1997) Qualitative Research in Information Systems. MIS Quarterly.

ORLIKOWSKI, W. & HOFMAN, J. (1997) An Improvisational Model of Change Management: The Case of Groupware Technologies. *Sloan Management Review,* 28**,** 11-21.

ORLIKOWSKI, W. J. (1993) CASE Tools as Organizational Change: Investigating Incremental and Radical Changes in Systems Development. *MIS Quarterly,* 17**,** 309.

SAHAY, S. & AVGEROU, C. (2002) Introducing the Special Issue on Information and Communication Technologies in Developing Countries. *The Information Society,* 18**,** 73-76.

SCOTT, J. E. & VESSEY, I. (2002) Managing risks in enterprise systems implementations. *Communications of the ACM,* 45 74 - 81

WALSHAM, G., ROBEY, D. & SAHAY, S. (2007) Special Issue on Information Systems in Developing Countries. *MIS Quarterly,* 31**,** 317.

WALSHAM, G. & SAHAY, S. (2006) Research on Information Systems in Developing Countries: Current Landscape and Future Prospects. *Information Technology for Development,* 12**,** 7-24.

YIN, R. K. (2003) *Case Study Research - Design and Methods,* California, Sage Publications, Inc.

YOO, S.-H. (2003) Does information technology contribute to economic growth in developing countries? a cross-country analysis. *Applied Economics Letters,* 10**,** 679 - 682.

ZMUD, R. W. & COX, J. F. (1979) The Implementation Process: A Change Approach. *MIS Quarterly,* 3**,** 35-43.