REQUIREMENTS FOR A SOFTWARE PROCESS MAINTENANCE FRAMEWORK FOR EXECUTIVE INFORMATION SYSTEMS IN THE TELECOMMUNICATIONS INDUSTRY

Nalinpat PORRAWATPREYAKORN¹, ¹ University of Vienna, Liebiggasse 4, A – 1010 Vienna, Austria, a0848231@unet.univie.ac.at, Gerald.Quirchmayr@univie.ac.at

Gerald QUIRCHMAYR¹, ² University of South Australia, Mawson Lakes, SA 5095, Australia, Gerald.Quirchmayr@unisa.edu.au

Wichian CHUTIMASKUL, King Mongkut’s University of Technology Thonburi, 126 Prachauthit Rd., Tungkru, Bangkok, 10140, Thailand, wichian@sit.kmutt.ac.th

ABSTRACT

This paper presents the interview findings of the current situation in software development for Executive Information Systems (EIS) in the Thai telecommunications industry and identifies critical success factors (CSFs) that can serve as requisite requirements for a software process maintenance framework in this context. The results show that software development teams do not perceive existing formal routines as an efficient way to manage software development processes and eventually to deliver quality results. One of the factors that play a central role in quality software development is an efficient software process. Efficiency requires project management activities to enable the proper task execution. Unfortunately, existing agile methods, e.g., Scrum and XP offer only inadequate support for project management, e.g., limited support for subcontracting and developing software that demands high level of quality control. For continuously efficient software processes, this study thus proposes a framework aimed at the establishment and improvement of software development processes covering adequate software engineering and project management perspectives. The overall goal of the resulting framework therefore is to contribute to the improvement of software process development.

KEY-WORDS: Software Development, Project Management, Software Process Improvement, Software Process Maintenance

1. MOTIVATION AND BACKGROUND

The best known traditional software development approach still is the waterfall approach, which in fact is the oldest original approach. It is a systematic and sequential pattern reaching from an initial feasibility study to the maintenance of the developed IS. However, there are several limitations, e.g., well-defined requirements, time-consuming, too much documentation and high cost (Jirachiefpattana, 1996). Agile methods, e.g., eXtreme programming (XP), Scrum, Dynamic Systems Development Method (DSDM) and Feature Driven Development (FDD), were thus developed to overcome these limitations. They are gaining recognition in the software development community due to their response to market expectation, i.e., innovative and high quality software (Highsmith & Cockburn, 2001). Moreover, CSFs for successful agile software development are identified by a multitude of studies (Attarzadeh & Ow, 2008; Ceschi et al., 2005; Chow & Cao, 2008; Gemuenden & Lechler, 1997). However, software development methods should be efficient (Kumar & Welke 1992). Efficiency requires project management activities to enable the proper execution of software development tasks. Project management thus provides the backbone for efficient software development (Abrahamsson et al., 2003; Gilb, 1988). From this view, some agile methods (e.g., Scrum, DSDM and FDD) are supplemented with guidelines on project management that allow for rapid delivery of quality products. Nevertheless, there generally is no comprehensive project management support (Abrahamsson et al., 2003). Scrum which is definitely the most popular (Shalloway & Trott, 2009) offers limited project management support, e.g., for subcontracting, developing with large teams, developing...
software that demands high quality control, and distributed development environments (Highsmith & Cockburn, 2001; Ionel, 2008; Shalloway et al., 2009; Turk et al., 2002). Albeit, researchers such as Turk et al. (2002) suggest that traditional project management practices are an applicable way, so far no integrated method offering adequate project management support to overcome these Scrum’s limitations has been identified. PMBOK is the broadest and most widely used standard reference of industry best project management practices (Thomas & Tilke, 2009) and definitely compatible with agile ways (PMI, 2008). Hence, there is not a great need but also a great opportunity to develop an integrated PMBOK and Scrum approach.

Besides, the quality of the software development process results in the quality of software (Humphrey, 1989). A software development process generally deals with how it can be implemented, but not so much with what processes should be implemented. Thus, only “how” cannot guarantee that software quality will be delivered. Software process improvement (SPI) can produce the quality of the software development process (Huang et al., 2005) that results in software quality (Lehman, 1991). CMMI is a well-accepted model for improving the performance of software development processes and software quality, and referring to what processes should be implemented to achieve successful software development (SEI, 2002). Hence, CMMI is considered as an effective way to maintain software process for continuous efficiency.

Motivated by the situation described above, this paper thus proposes a software process maintenance framework, which in this context means efficient software process establishment and improvement. This paper also focuses on EIS development in the Thai telecommunications industry as a case study. Recently, the use of EIS has significantly increased since the success of EIS in developed countries stimulates a number of executives to adapt this Information System (IS) into their organizations in order to compete in an increasingly competitive environment. EIS are different from Transaction Processing Systems (TPS), Management Information Systems (MIS), and Decision Support Systems (DSS) in terms of problems addressed, users and data used. TPS serve operational management by performing and recording the daily routine transactions necessary to conduct the business and solve structured problems which have standard solutions. Both MIS and DSS provide middle management. However, there are different characteristics for the way in which MIS deal with summarized and compressed data from the TPS and sometimes does an analysis of that summarized data to solve structured problems, while DSS use data from TPS, MIS, and external sources to solve semi-structured problems which only part of the problem has a structured quality. EIS provide information for top management to solve unstructured problems which have no standard solutions for resolving the situation, so that they can identify problems and opportunities by combining internal and external information that is relevant to decision making (Jirachiefpattana, 1996; Laudon & Laudon, 2009). EIS can directly aid and support communications, coordination, planning and control functions of managers and executives in an organization. Supporting this, Nord & Nord (1995) argue that utilizing EIS software can provide valuable benefits, i.e., better communication, increased confidence in decision making, and eventually increased profits. Moreover, Telecommunications still is one of the most rapidly evolving competitive markets and one of the fastest-growing areas of technology in the world. Thailand’s telecommunications sector is worth mentioning that it has continued to grow, in the last five years. It generated 8.1 billion dollars in service revenue in 2008 (Pyramid Research, 2009). The main engine of growth is a broadband service that its growth rate was more than 700% from 2003 to 2004 (The World Bank, 2008). Lately reported, Thailand was one of five Asian countries ranked among the world’s top ten fastest-growing consumer broadband markets in 2008 (Point Topic, 2009).

EIS require fundamental revision and software development methods that must be able to deal with rapid evolution (Jirachiefpattana et al., 1996). Unfortunately, EIS is likely to be more difficult in Thailand due to difficult software development environments, e.g., organizational culture, lack of user participation and inappropriate methods (Jirachiefpattana et al., 1996). “Do the problems identified in prior research still exist in EIS development in the Thai context? Do the problems involve both software engineering and project management aspects? What are factors affecting successful EIS development?” are three fundamental questions of this study. The main methods used during this research are qualitative analysis via interviews and literature review methods. For the interviews, two organizations were identified and are thought to representative of EIS in the Thai telecommunications industry. Field data collection was performed during March and April 2009. Questions about the software development process were developed for project managers, developers and coordinators. These questions were answered in structured interviews that ranged in length from one to two hours.
2. A LOOK AT THE CURRENT SITUATION IN EIS DEVELOPMENT IN THE THAI TELECOMMUNICATIONS INDUSTRY

For getting an idea of the current situation in the EIS development in the Thai telecommunications (by focusing on Internet services), we use findings of interviews with software development teams working for two companies: True Corporation Public Company Limited and TOT Public Company Limited. These companies are the two biggest broadband Internet Service Providers (ISP) in the Bangkok region and have their own optical fiber cable networks in Bangkok and in the vicinity. Even though there are many ISP in Thailand, most of them still lease bandwidth from one of these two companies. With 85% True Corporate also has the largest market share (Thailand Guru, 2009; The World Bank, 2008). However, the size of companies does not affect the model of EIS development. In the organizational context of EIS implementation, the results reveal that the executives could sometimes not provide adequate participation in the projects. Subordinates did not have full authorities when it came to making decisions. Communication processes during EIS in organizations also were quite complicated, e.g., executives or users do neither have good cooperation nor do they participate well. These limitations resulted in development teams sometimes not being able to identify the information requirements from executives effectively, often having to wait for Steering Committee decisions, and resulted in an extensive organizational process. As was to be expected, the projects were delayed.

Given the underlying EIS development strategies of prototyping and outsourcing, the EIS development project with a small team in True Corporation had a short duration. True Corporation used a prototyping model. The development process involved requirements analysis, preliminary design, prototype design, construction and testing, implementation and maintenance. In the other case, the EIS development project with a big team in TOT had an initial period of two years. Outsourcing usually covers a wide range of contractual arrangements ranging form contract programmers to third party facilities management (Lacity & Willcocks, 2001). The EIS development in TOT was to some extent outsourced. One of reasons for employing the consultant was that the internal staff lacked knowledge and experience in EIS development. Although the development methodology used terms like prototyping and module delivery, it can best be characterized as a variant of the waterfall approach. The development process involved a large execution of requirements analysis, system development, user acceptance, system installation, and maintenance. During EIS development, the teams face similar problems. For example, users provide inadequate requirements specifications and quite frequently change their requirements; users have only limited Information Technology (IT)/IS skills; the teams do sometimes not have enough training support; and so on. This situation is quite typical, and not limited to the Thai telecommunications sector.

Besides, data quality is primarily concerned with the data reliability (Starreveld et al., 1976; 1994) and affects the software reliability. Reliability consists of three dimensions: accuracy (i.e., consistent and accurate data), completeness (i.e., complete data), and timeliness (i.e., up to date data) (Starreveld et al., 1994). Data are of high quality if they are fit for users’ intended uses (Juran & Godfrey, 1999). Data quality can thus determine success or failure of software development. Moreover, successful software development should consider internal features (i.e., stakeholders and policy, and development methodology) and external features (i.e., information and communication technology (ICT), and the environment) (Chutimaskul et al., 2008). The stakeholders and policy feature is referred to the quality of organization and people. The development methodology feature is referred to the quality of software processes. The quality of ICT feature must consider ICT competency, vendor support, and ICT personal, whilst the quality of environment feature must consider external factors, i.e., requirement volatility. The findings show that the EIS problems in the Thai context identified in prior research still exist, and these problems mainly involve software engineering and project management aspects. Based on the mentioned quality aspects the problems/failures of the findings can be summarized in Table 1.
<table>
<thead>
<tr>
<th>Dimension</th>
<th>Failure Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization</td>
<td>Lack of management commitment, organizational culture too traditional and political, lack of agile logistical arrangement</td>
</tr>
<tr>
<td>People</td>
<td>Lack of necessary skill-set, lack of project management competence, lack of good user participation and cooperation, lack of teamwork.</td>
</tr>
<tr>
<td>Process</td>
<td>Ill-defined project scope, requirements, and planning, user having no full authority</td>
</tr>
<tr>
<td>Technology</td>
<td>Lack of provision and support of training to team, inappropriateness of methods and tools</td>
</tr>
<tr>
<td>Project</td>
<td>Team size</td>
</tr>
<tr>
<td>Data</td>
<td>Inconsistent data, contradictory data, redundant data, missing data, out of date data</td>
</tr>
</tbody>
</table>

**TABLE 1 - The failure factors in EIS development**

3. SOME OF THE RELEVANT ISSUES CURRENTLY DISCUSSED IN EUROPE

Since many European developed countries such as the UK, France, Nordic and Germanic countries are recognized for success in EIS and software development, this study focuses on software development with CSFs, currently both traditional project management and agile software development approaches are gaining great popularity in the software development sector (Griffiths, 2004). PMBOK and Scrum are definitely the most popular for project management and agile software projects, respectively (Shalloway & Trott, 2009; Thomas & Tilke, 2009). PMBOK nevertheless has been influenced by agile tendency as we can see that the latest PMBOK edition promotes its practices in an agile way (PMI, 2008). Scrum is management-oriented and has more advantages in facets of responsiveness to environment, team flexibility and creativity, knowledge transfer during project, and high probability of success (Schwaber, 1995). However, it offers limited support for project management, i.e., limited support for scope, time, cost, risk, quality, procurement and documentation management (Ionel, 2008; Turk et al., 2002). An efficient software process needs to be able to cope with project and process management activities.

To provide adequate support for these two aspects, an integrated PMBOK and Scrum approach is thus taken into account. Supporting this, Fitsilis (2008) suggests that connecting Scrum with PMBOK can benefit the software development teams since processes in Scrum and PMBOK, are addressed in a compatible way. For understanding telecommunication business process requirements, eTOM is also considered. It has also proven to be a successful framework (PRLog, 2009). In the last few years several successful implementations of eTOM have been reported in a number of papers, e.g., the case studies by Forsyth et al. (2008), Interfacing (2008), and PRLog (2009).

This study also intends to maintain the software process to continuously be efficient by assessment and improvement. CMMI is widely adopted appraisal approach that helps improve software process, product quality and project reliability, and eliminate problem and defect causes (Persse, 2001). Currently, many organizations are increasingly interested in adopting CMMI with agile methods together (Marçal et al., 2008). There are evidences that CMMI and agile, especially Scrum, can considerably coexist (Diaz et al., 2009; Jakobsen & Sutherland, 2009; Marçal et al., 2008; Sutherland et al., 2007). CMMI is thus deemed for this study. Besides, defined CFSs should be oriented towards completing software development effectively. Based on Chan & Thong (2009) and prior studies on the factors that impact on agile software projects, these identified factors can be summarized in Table 2.
<table>
<thead>
<tr>
<th>Source</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ceschi et al. (2005)</td>
<td>Teamwork, individual ability, motivation</td>
</tr>
<tr>
<td>Chow &amp; Cao (2008)</td>
<td>Management commitment, organizational environment, team environment,</td>
</tr>
<tr>
<td></td>
<td>team capability, user involvement, project management process, project</td>
</tr>
<tr>
<td></td>
<td>definition process, agile software engineering techniques, delivery</td>
</tr>
<tr>
<td></td>
<td>strategy, project nature, project type, project schedule</td>
</tr>
<tr>
<td>Cockburn &amp; Highsmith (2001)</td>
<td>Individual competence, management support, communication,</td>
</tr>
<tr>
<td></td>
<td>compatibility of agile methods, teamwork, project type, team size</td>
</tr>
<tr>
<td>Colm &amp; Ford (2003)</td>
<td>Resistance due to past experience, leadership, career consequences,</td>
</tr>
<tr>
<td></td>
<td>developer competence</td>
</tr>
<tr>
<td>McManus (2003)</td>
<td>Individual competence, compatibility between skills and tasks, good</td>
</tr>
<tr>
<td></td>
<td>communication skills, experience in software development</td>
</tr>
<tr>
<td>Nerur et al. (2005)</td>
<td>Organizational culture, management style, organizational form, management of software development knowledge, reward systems, teamwork, competence, user relationships, existing technology and tools, training</td>
</tr>
<tr>
<td>Schatz &amp; Abdelshafi (2005)</td>
<td>External support, teamwork, compatibility of methods, negotiation skills</td>
</tr>
</tbody>
</table>

**TABLE 2 -** A summary of the identified factors of agile software projects

There is an amount of IS research that focuses on the improvement of software development success in aspects of speed, effectiveness, efficacy, and low cost, to only name the most important ones. It is commonly accepted that no single method can serve for all types of software projects and all types of project objectives. In sum, this study deems agile software process, SPI, and CSFs for a successful software process maintenance framework for software development in the Thai telecommunications sector. The setting of this focus was also determined by the scholarship supporting this work.

4. FOUNDATIONS OF OUR STUDY – WHERE WE CAN START FROM

To cope with the EIS problem/failure factors and answer the research question of “What are factors affecting successful EIS development?”, the following two sets of principles and three models are used as the basis for proposing a software process maintenance framework, which consists of two main parts. For the first part of software process establishment, the two sets of principles are derived from the core of the PMBOK and the core of the eTOM, which the authors then try to merge into the Scrum model. For the second part of SPI, CMMI and CSF approaches are employed. A number of prior studies concur that a successful software process should be viewed in terms of CSFs rather than key process areas (Fitzgerald & O’Kane, 1999; Niazi et al., 2004; 2005a; 2005b; Somers & Nelson 2001). These studies emphasize the use and the importance of the CSF approach in SPI and have confirmed the value of the CSF approach in the field of information technology (Huotari & Wilson, 2001; Khandelwal & Ferguson, 1999; Niazi et al., 2004; 2005a; 2005b; Pellow & Wilson, 1993; Somers & Nelson, 2001; Tyran & George, 1993). Moreover, a multitude of studies cite the “reviews” factor that has a major impact on successful SPI and some of these studies also cite that it is only one factor corresponding to the top CMMI-based maturity level (Niazi et al., 2003; 2004; 2005a; 2005b; Somers & Nelson, 2001). Therefore, the “reviews” factor is used as an additional key requirement for the proposed framework. In addition, quality affects software use and user satisfaction (DeLone & McLean, 2003). TAM provides a basis for understanding intention to use, software use, and user satisfaction by two factors (i.e., perceived usefulness and perceived ease of use). Thus, these two factors of TAM are used as additional key requirements for the proposed framework.

4.1 Principles of the Project Management Body of Knowledge Guide (PMBOK)

PMBOK developed by PMI (PMI, 2008) is process-based. The processes are described in terms of inputs, tools and techniques, and outputs. The guide recognizes 42 processes that fall into five basic process groups (i.e., initiating, planning, executing, controlling, and closing) and nine knowledge areas (i.e., integration, scope, time, cost, quality, human resource, communications, risk, and procurement) that are typical of almost all projects. PMBOK provides a concise summary of and reference to generally accepted project management principles that can easily be integrated in other approaches.
4.2 Principles of the Enhanced Telecom Operations Map (eTOM)

The Business Process Framework, known as eTOM, is a widely deployed and accepted standard for business processes in the telecommunications industry. The eTOM represents the whole of a Service Provider/Operator's enterprise environment in a hierarchy of process elements that capture process detail at various levels. It provides a basis for understanding and managing portfolios of IT applications in terms of business process requirements and enables the creation of consistent and high-quality end-to-end process flows, with opportunities for cost and performance improvement, and for the reuse of existing processes and systems (TM Forum, 2009).

4.3 Scrum Model

Scrum, originally developed by Ken Schwaber is an iterative, incremental process of software development. Scrum focuses on project management in situations where it is difficult to plan ahead and where feedback loops constitute the core element. Software is developed by a self-organizing team in increments or sprints, starting with planning and ending with a review. Features to be implemented are registered in a backlog. Then, the product owner decides which backlog items should be developed in the following sprint. Team members coordinate their work in a daily stand-up meeting. The scrum master is in charge of solving problems that stop the team from working effectively (Schwaber & Beedle, 2001).

4.4 Capability Maturity Model Integration Approach (CMMI)

CMMI (SEI, 2002) is a widely known appraisal approach for continuous SPI. It consists of best practices that address development and maintenance activities covering the software lifecycle. CMMI itself has two representations: staged and continuous. The staged representation focuses on process areas organized by maturity levels, while in continuous representation each process area is rated in terms of capability level. These processes refer to “what to do” to achieve successful software development. CMMI strives to achieve process consistency, predictability and reliability.

4.5 Technology Acceptance Model (TAM)

The Technology Acceptance Model (TAM) by Fred D. Davis (1986) is an influential extension of Ajzen and Fishbein’s theory of reasoned action (TRA). TAM is a model derived from a theory that addresses the issue of how users come to accept and use a technology. The two specific variables of perceived usefulness (PU) and perceived ease of use (PEOU) are hypothesized to be fundamental determinants of user acceptance of any technology, which should not be ignored by IT/IS development teams attempting to design or implement successful systems. TAM has also been applied in numerous studies testing user acceptance of IT/IS, e.g., word processors, spreadsheet applications, e-mail, web browser, telemedicine, and websites (Gao, 2005).

Proper framework of software process maintenance is vital to achievement of effective software development success. To build such guidance, the requirements which are specific to the EIS development in the Thai telecommunications sector thus need to be identified.

5. REQUIREMENTS FOR A SUCCESSFUL SOFTWARE PROCESS MAINTENANCE FRAMEWORK

Stating requirements is very important for the design of all mechanisms. Requirements for the successful implementation of the proposed software process maintenance framework are summarized into Table 3, which are also compared to CSFs of prior studies on agile software projects. This is based on the PU and PEOU factors of TAM, and the consolidation of a number of failure/success factors listed in Tables 1 and 2 which share similar characteristics. This also answers the research question of “What are factors affecting successful EIS development?”
6. TOWARDS THE PROPOSED SOFTWARE PROCESS MAINTENANCE FRAMEWORK BASED ON THE IDENTIFIED REQUIREMENTS

Software developers are still under pressure to deliver quality results due to limitations of methods employed and the influence of various factors. Albeit management-oriented agile methods are available, they provide inadequate support suitable for EIS development in the Thai context, i.e., the demand of high quality control and procurement management for IT/IS outsourcing. To overcome this problem, a framework for establishment and improvement of an efficient software development process is needed. In general, according to (Wallin & Crnkovic, 2003) each software development project is run through a platform deployment lifecycle of four stages: ideas, feasibility study, software development, and rollout. First, the ideas stage starts with a collection of ideas for end user solutions that can be enable through the new software platform. Second, the feasibility study stage is to compile the information needed for the responsible management to make a decision whether to start a pilot development project. Third, the software development stage is where the approved pilot development project is run based on the feasibility study results. Last, the rollout stage runs when developed software is ready to be employed. For this platform deployment lifecycle, most development theories have similar methods for the stages of ideas, feasibility study, and rollout. Except for the utilization of software development methodologies, it depends on the type, nature, and characteristics of each project. To be clear, this study mainly focuses on the software development stage as presented in Figure 1.

![FIGURE 1 - The primary focus of this study](image-url)
Because the proposed software process maintenance framework aims to provide engineering and management directions to achieve better performance of software process development, the PMBOK, eTOM, Scrum, CMMI, and TAM models all have strong benefits for the proposed framework. In the framework, there are two main parts. First, for establishing software processes the Scrum and PMBOK processes are mapped into five iterative process groups: initiating, planning, executing, controlling, and closing. eTOM is also mapped together by focusing on the top-down principle for analysis of the impact of required new process against existing business processes and the bottom-up principle for analysis of the data available from the network. Second, for assessing and improving the processes the staged representation of CMMI and CSF approaches are employed. At this stage of this study, four CMMI maturity levels of initial, managed, defined, and optimizing are adopted for a software development maturity model of this study. The main reason not to replicate the CMMI maturity level-4 “quantitatively managed” is that the two key CMMI practices of (1) establishment and maintenance of quantitative objectives for the process and (2) stabilization of the performance of one or more sub-processes to determine its ability to achieve are not compatible with agile best practices (Jain, 2002). Additionally, there is no success factor cited in literature that directly relates to this level (Niazi et al., 2004; 2005a; 2005b). Consequently, the maturity levels 1 though 4 of the model are initial, managed, defined, and optimizing, respectively.

Non-identified requirements or other requirements unidentified in the section 5 that affect successful EIS development might be discovered in the future. To be clear, at this stage the identified requirements of this work are used as the identified CSFs. The identified CSFs were categorized into their corresponding maturity levels. There is no category for the level-1 “initial” since this level does not have to be achieved due to its chaotic characteristic. Similarly, CMMI does not have process areas for this level. The level-2 “managed” contains CSFs supporting project management activities which are the foundation for all subsequent levels. This level contains management commitment, project management process, project type, user involvement, and training support. The level-3 “defined” contains CSFs that support to design systematic structures for SPI implementation, i.e., team environment, organizational environment, agile software engineering process, appropriate methods and tools, team capability, team size, and data quality. The level-4 “optimizing” contains the “reviews” factor to support continuous SPI activities. We considered categories of CSFs regarding SPI implementation process of prior empirical studies (Niazi et al., 2004; 2005a; 2005b) in order to have more confidence in the categorization of these CSFs. Besides, the CSFs of PU and PEOU are utilized for measurement of software quality, and intention to software use. Figure 2 shows the conceptual framework of this work.

FIGURE 2 - The proposed conceptual software process maintenance framework

The key requirements identified for the proposed software process framework serve as basis for designing an abstract level model of the proposed conceptual framework. The next step will be a more detailed process mapping between PMBOK, eTOM, and Scrum and the more detailed relationship description between CSFs and CMMI-based software development maturity levels.
7. CONCLUSION

In this contribution, the interview findings show that software developers do not perceive formal routines as an efficient way to manage development and management processes. It means that problems defined in prior research on EIS development in Thailand still exist today. The factor that plays a major role in software development success is an effective software process. This paper therefore proposes a software process maintenance framework, which in this context means efficient software process establishment and improvement. The first step to construct the framework is to identify the CSFs as key requirements that affect successful EIS development in the Thai telecommunication industry. To deal with the problems, two principle sets of the PMBOK and eTOM and three models of CMMI, Scrum, and TAM all serve as inspirational principles and models for the proposed software process maintenance framework. PMBOK provides general guidance covering all facets of project management in the traditional sense. eTOM fulfills specific requirements in the area of the telecommunications industry. Scrum is commonly used in the agile software development context. CMMI is an SPI approach that provides organizations with the essential elements of effective processes that help improve their performance. The PU and PEOU of TAM are significant factors for the technology acceptance side of the framework. The framework is at this stage only aimed at contributing to the improvement of software process development in the Thai telecommunications sector, but there are indications that some of the major findings of this research might actually be of a more general nature and hence of a wider applicability. Accordingly the framework could be employed for other kinds of IS systems and/or other business sectors by replacing the eTOM-related part of the framework with suitable domain-specific business process guidelines. The next steps of this work are (1) the detailed process mapping between PMBOK, eTOM, and Scrum, and (2) the more detailed relationship description between CSFs and CMMI-based software development maturity levels. Practical tests of our approach will be carried out in cooperation with companies in the telecommunications sector in Thailand in the last quarter of 2010.

Acknowledgments. The authors would like to thank ÖAD, the Austrian Agency for International Cooperation in Education and Research, and the Higher Education Commission of Thailand for supporting this work in the form of a scholarship for Nalipat Porrawatpreyakorn. The authors would also like to acknowledge the valuable suggestions made by Professor Somchai Chatvichienchai, Phil Robinson, and colleagues from KMUTT.

REFERENCES


Copyright© Institut Fidal Inc (2010)