การจัดการความเสี่ยงในการจัดซื้อซอฟต์แวร์
กรณีศึกษาบริษัทในประเทศไทย

นาย เศรษฐภัทร เย็นเยือก

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และสารสนเทศ ภาควิชาคณิตศาสตร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

RISK MANAGEMENT IN SOFTWARE PURCHASING: AN EMPIRICAL STUDY OF THAI'S COMPANIES

Mr. Setaphat Yenyuak

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science and Information
Department of Mathematics
Faculty of Science
Chulalongkorn University
Academic Year 2008

Thesis Title	Risk Management in Software Purchasing : An Empirical Study of Thai's Company
By	Setaphat Yenyuak
Field of Study	Computer Science and Information
Advisor	Assistant Professor Pattarasinee Bhattarakosol, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

S. Hannongbua .... Dean of the Faculty of Science

(Professor Supot Hannongbua, Dr.rer.nat)

THESIS COMMITTEE

C. Lur .... Chairman

(Professor Chidchanok Lursinsap, Ph.D.)

P. Bhattarakosol .... Advisor

(Assistant Professor Pattarasinee Bhattarakosol, Ph.D.)

.... Examiner

(Assistant Professor Nagul Cooharojananone, Ph.D.)

Sunee Pongpinigpinyo .... External Examiner

(Sunee Pongpinigpinyo, Ph.D.)

เศรษฐภัทร เย็นเยือก : การจัดการความเสี่ยงในการจัดซื้อซอฟต์แวร์ กรณีศึกษาบริษัทในประเทศไทย. (RISK MANAGEMENT IN SOFTWARE PURCHASING: AN EMPIRICAL STUDY OF THAI'S COMPANIES) **ผศ. ดร. ภัทรสินี ภัทรโกศล:** 92 **หน้า.**

ปัจจุบันการแข่งขันในโลกธุรกิจเป็นไปอย่างเข้มข้น ซอฟต์แวร์เดิมไม่สามารถรองรับความต้องการทางธุรกิจที่มีการเปลี่ยนแปลงตลอดเวลาได้ การจัดซื้อซอฟต์แวร์เป็นทางออกในการแก้ปัญหาเพื่อช่วยให้องค์กรตอบสนองความต้องการได้ บางองค์กรประสบความสำเร็จในการจัดซื้อแต่บางองค์กรก็ประสบความล้มเหลว เนื่องจากมีความแตกต่างในกระบวนการจัดการความเสี่ยงที่ถูกเลือกใช้ในการเลือกซื้อซอฟต์แวร์ของแต่ละองค์กร ดังนั้นจึงทำการสำรวจกระบวนการจัดการความเสี่ยงในหลากหลายประเภทธุรกิจของแต่ละองค์กรในประเทศไทย, สรุปผลและนำเสนอกระบวนการจัดการความเสี่ยงที่มีความแม่นยำถูกต้องมากที่สุด เพื่อช่วยลดปัญหาความล้มเหลวเปรียบเทียบกับความพึงพอใจของผู้ใช้ จากผลลัพธ์แสดงให้เห็นว่ากระบวนการจัดการความเสี่ยงมาตรฐานไม่สามารถตอบสนองความพึงพอใจของผู้ใช้ได้เพียงพอ ในขณะที่กระบวนการจัดการความเสี่ยงที่นำเสนอให้ผลลัพธ์ที่ดีกว่า

ภาควิชา คณิตศาสตร์ ลายมือชื่อนิสิต เศรษฐภัทร เย็นเยือก
สาขาวิชา วิทยาการคอมพิวเตอร์และสารสนเทศ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก P. Bhattarakosol
ปีการศึกษา 2551

497 36092 23: MAJOR COMPUTER SCIENCE AND INFORMATION
KEYWORDS : USER SATISFACTION RATE / RISK MANAGEMENT PROCESS / SOFTWARE PURCHASING PROCESS

SETAPHAT YENYUAK: RISK MANAGEMENT IN SOFTWARE PURCHASING: AN EMPIRICAL STUDY OF THAI'S COMPANIES. ADVISOR: ASSISTANT PROFESSOR PATTARASINEE BHATTARAKOSOL, 92 pp.

When business competition is increased, it is possible that legacy software could not serve the response on business requirements. Software purchasing is the solution that could help organizations to take action on dynamically requirements. Some organizations have been successfully implemented new software while some fail according to the differences of risk management processes that selected from each organization. Therefore, a survey of risk management processes in various business organizations in Thailand had been performed and proposed the most accurately risk management procedure which are proof of concept and product prototype analysis that could reduce percentage of failure comparing with organizations' satisfaction rate. Additionally, the experimental results have shown that applying standard risk management process do not meet user satisfaction while proposed risk management process of Thai's organizations performed better results rather than standard.

Department : Mathematic Student's Signature
Field of Study : Computer Science and Information Advisor's Signature P. Bhattarakosol
Academic Year : 2551

Acknowledgements

I would like to express my gratitude to all those who made it possible for me to complete this thesis.

To Assistant Professor Dr. Pattarasinee Bhattarakosol, who has taught me and advised me during bachelor's degree and master's degree, I truly thank for her help, support and encouragement. All these things she provides to me are invaluable. I feel much honor to be her disciple.

Thanks for assist, encouragement, and support from all my friends both in the master program and in Bangkok Bank Company Limited.

Finally, I cannot express enough gratitude to my family – my beloved parents - for their love, and contribute. They have always supported and encouraged me to do my best in all the matters of my life.

Contents

page

List of Tables

List of Figures

CHAPTER I

INTRODUCTION

1.1 Background and Motivation

Since, there is no standard for purchasing software process, many organizations have defined their own processes to filter software that meet their requirements. According to these, a number of organizations fail to obtain their software as needed. Sometimes the purchased software could not properly fit their requirements, some features cannot be used, incompatible with their existing systems, or even do not meet user satisfaction [1][2][3]. Thus, various risk management processes have been proposed to protect and prevent the failure in the software purchasing.

There are several risk management processes to apply for software selection, such as proof of concept [36][37][41], requirement analysis [30][31][32][33], product prototype analysis [38][39][40], software presentation [30], software demonstration [30], and learning software behaviors and vendors' services from other organizations. However, combinations of these methods could reduce the risk. Numerous studies have shown that over half of software development projects fail; the significant reason that causes the failure is breakdown to the "requirement elicitation" process. Additionally, problems occurred during software implementation are, such as vendor services, maintenance costs, features and functions do not meet the clients' demands, incompatible with the legacy system, fail user satisfaction and software liabilities, etc.

Referring to the requirement elicitation process, this process produces requirements of users that lead to the software purchasing. Therefore, the details obtained from this process are very significantly important. Since there are various tools to be applied in this elicitation process, such as use cases, and data flow diagrams, etc., these tools must be properly implemented by the software developer team, especially software analysts. Unfortunately, the final results that are the delivered software or purchased software, mostly, do not completely meet the needs of the entire organization.

Although there is failure in software purchasing in some organizations, applying risk management helps reducing this failure in other organizations as well. Therefore, this research focuses in the user satisfaction, results of Thai companies on each risk management. The expected outcome from this research is the proper procedures in software purchasing that every organization can apply.

Since the standard software purchasing procedure does not meet the users' expectation, organizations have to find their solutions that are much reliable and cost effectiveness. Generally, there are different risk management processes that apply on each company. Unfortunately, there are not all companies that would success in their software purchasing procedures. This research tries to reduce software purchasing failure rate by studying common six risk management methods: requirement analysis, product prototype analysis, proof of concept, software demonstration, software presentation and learning software for other organization.

Requirement Analysis is used as baselines to compare satisfaction rate with each other procedures. Requirement analysis is the process of determining users' expectations for a newly software. It covers complex tasks of eliciting, modeling, analyzing, and documenting the requirements of all users [34]. The outcomes from this process are the solutions for the system and software designing phase. Thus, using an inappropriate tool leads to the wrong decision of software development direction which has a main effect to the strength of the organization [35].

In some organizations, the product prototype analysis method is applied instead of performing the direct requirement analysis process. So, requirements can be captured easily from uneducated computer users. The objectives of the prototyping are used for determining software features and function comparing which legacy software. It will show software concept and advantage. The results are the solution that can cover all specifications of new features.

According to the previous risk management method perform under a user acceptance levels, the requirement analysis and the product prototype analysis methods, another solution in purchasing software is the proof of concept. This process mostly concerns in timeline and quality of the delivered software which cannot be tracked by the previous methods. Proof of concept is a set of proofs or tests defined by organizations for software specifications[36][37]. It covers many tasks to be performed, including defining success criteria, extracting requirements, define timeline, and proof of software functions. The results confirm that the purchasing software can be implemented within the success criteria.

Many organizations try to apply previous processes to select or classify appropriated software. There is another process that is similar to the product prototype analysis, called software demonstration. This process allows users to examine the purchasing software before making decision. This process demonstrates software behavior, interfaces, features and functions, which can prove that the software is user friendly and compatibility to the existing system.

The different way to analyze the purchasing software is the software presentation method. This process is less complexity than other processes. It introduces software advantages, features, and functions. It is also similar to the software demonstration procedure except that the organizations cannot run the purchasing software. The outcome is that organization could understand software product, features, and functions without experiencing the system.

Although there are various methods mentioned above, there is another alternative which is the easiest way to make decision in purchasing software. This method is to obtain details of the required software by learning its behavior and vendor services from other organizations. The advantage of this method is that the organizations spend the lowest cost to obtain software information than other processes. According to this process, organizations can recognize vendors’ reputation and reliable services. Thus, vendors with bad reputation can be eliminated. Consequently, organizations can avoid disasters during implementing the purchased software.

Referring to all methods mention previously, the requirements of users still are not completely served. Therefore, this studied shows that using combination between two risk management processes as the baseline can obtain a better result than applying only one risk management procedure.

In order to prove the assumption stated above, the objective of this Dissertation is to introduce a software purchasing process that reduces risk and preciously meet users' requirements.

1.2 Objective

The objectives of this thesis are as follows.

1. Prove the correctness of the combination of two software processing methods.
2. Propose the most suitable software purchasing process that reduces risk in software purchasing and increases users' satisfaction, including the implementation time.

1.3 Scope of Thesis and constraint

As the fact that current business organizations are relied on efficient and effective software, unfortunately, most purchased software cannot serve organizations as needed according to some error during the software purchasing process. However, this study does not focus in the software purchasing of the government section because the damage that caused by invalid software delivery is not critical as occurring in the business section.

Although, there are various methods were introduced and implemented from each organization, none of them can completely prevent the failure. As a result, this research focuses on the software purchasing process problem for the business area. However, the solution obtained from this research can apply not only the business careers, but also various group.

1.4 Definition

User Satisfaction rate: The percentage of users' acceptance level comes from analyze collected data. It identify user attitude with purchasing software.

Risk Management Process: The procedures that assess, identify, mitigate, and monitor risk.

1.5 Benefits

According to the result of this thesis, a software purchasing process will be presented and the risk of software failure of the organizations is eliminated. Therefore, the organizations can perform their task with satisfaction and accuracy. Consequently, the business competition of the organization is increased.

1.6 Outline of this Thesis

The rest of this thesis is organized as follows. Chapter 2 introduces the standard risk management process which was used in the normal software purchasing process. In Chapter 3, the organization surveys method for collecting data from organizations, is elaborated; data analysis and the results are described in Chapter 4. Finally, discussion and conclusion about standard risk management against Thai's risk management procedure have been drawn in Chapter 5.

CHAPTER II

FUNDAMENTAL KNOWLEDGE AND LITERATURE REVIEW

There are several risk management methods that are generally applied in any business companies, such that requirement analysis, proof of concept (POC), software demonstration, software presentation, product prototype analysis, learning software behavior and vendor services from other organizations.

The standard risk management process [4] on software purchasing which is widely used in every IT-based organization is based on the requirement analysis process. The requirement analysis process is the process for understanding users' needs and expectations from a proposed system or applications. Additionally, this process is the most important stage that must be well-defined in the Software Development Life Cycle (SDLC) model, otherwise its failures affect to all other following processes, including the software failure to the entire organization.

2.1 What is the Risk Management?

Risk Management [5][6][7] in this context is the process of defining risks, sizing them and defining actions to deal with them.

Risk is traditionally considered to be a hazard; the risk of things going wrong [8]. A risk is any factors or events that could impact the organization's ability to meet its objectives. For example, the risk of losing implemented purchasing software, or the risk of useless purchasing software.

Risk needs a reference point to make the risk assessment process effective [9]. A risk needs to be detected. Many organizations have to define the possibility of each evident that might be harmful to overall processes [10]. The typical risks areas are as follows.

- Failures in business processes
- Loss of key people
- Actions by competitors
- Changes in user requirements
- New implements in technology with mostly concern in this paper

In reality, even though there is well planning but the trouble things may frequently occur. Thus, the competent and professional managers want to ensure that these risks have been evaluated, protected, and detected.

2.2 Software Risk Management

The hierarchy of Software Risk Management (SRM) methodologies discussed in [11] addresses two classes of functions: software purchasing acquisition, and software purchasing development. The basic methodological framework with which functions are managed is composed of supporting practices and constructs [12]. This framework for software risk management is supported by three groups of practices:

1. Software Risk Evaluation (SRE)
2. Continuous Risk Management (CRM)
3. Team Risk Management (TRM)

The complexity of software risk management cannot neither be understood nor appropriately addressed from the above methodological context alone [13]. To capture the multifarious aspects of this complexity, the hierarchical holographic modeling is introduced [14], where it can be considered two additional visions or dimensions: the temporal, and human dimensions. Thus, there are three dimensions that represent the holistic vision of the software risk management; these are the temporal dimension, the human dimension, and the methodological dimension.

The temporal dimension is decomposed into two sub-visions:

1. Macro vision represents the global perspective of the acquisition life cycle
2. Micro vision represents the view of the project manager.

The human dimension addresses the intellectual dimension of software acquisition which is the most critical dimension, since software development is such an intellectual activity. Four aspects are identified here:

1. Individual
2. Team
3. Management
4. Stakeholder (including customers and users)

The purpose of the methodological dimension [15] is to enable engineers, managers, and other decision makers to identify the risks associated with software acquisition, development, integration, and deployment. So, appropriate management and mitigation strategies can be developed on a timely basis [16]. Time is critical, and the risk detection must act early before a source of risk evolves into a major crisis. In other words, the heart of good risk management is mainly reactive in the risk migration and control rather than proactive in risk prevention and control [17]. Unfortunately, the reactive prevention cannot prevent the system failure, then ensuring the safe failure of the system must be the mandate of the software risk manager.

The values of the methodologies and tools for software risk management are many aspects. It supports organization for their wisely purchase the products and effectively and efficiently manage their projects. Furthermore, there are opportunities to continuously improve their processes [11].

It is important to note that the purchasing software risk methodologies have three fundamentally different objectives: risk prevention, risk mitigation and correction, and ensuring safe system failure. Since these objectives handle all risk that might impact to the project, the experience in these fundamental increases the completeness of productivity.

In order to achieve these three objectives, the following seven risk management principles are required [17]:

Shared product vision

Sharing product vision: this principle is based on a common purpose of software purchasing of the entire organization. The result of this process is to confirm that the purchasing software is perfectly suitable for their organizational functions. Additionally, this is also based on the collective commitment of decision making which guaranteed the results from discussion are correct. Furthermore, the shared product vision reduces chances of any failures that the organizations might miss their goals.

Teamwork

As the fact that every task must be performed by a teamwork, therefore, the success of a project also depend on this factor. Thus, good team members work cooperatively to achieve a common goal to reduce opportunity of risk in human dimension. Moreover, the organizations have to provide pooling talent, skills, and knowledge to improve performance of the team members.

Global perspective

One common factor of a software purchasing is that the Global perspective of the organization and software must be considered. The global perspective views software development within the context of the larger system-level definition, design, and development to ensure system correctness. The important of this perspective is that it recognizes both the potential value of opportunity and the potential impact of adverse effects, such as cost overrun, time delay, or failure to meet product specifications. This prevents risks that might occur in the system.

Forward-looking view

Not only the global perspective view to be considered, the developers also have to concern for the forward-looking view. This forward-looking view introduces project members to think toward tomorrow, identify uncertainties, and anticipate potential outcomes of the project which can predict the occurrence of the risks. The benefit of the forward-looking view is that it manages project resources and activities while anticipating uncertainties which can decrease the possibilities of risks.

Open communication

Another successive factor depends on the open communication among organizational staffs. The communication encourages the free flow of information between all project levels which every team member could experience this information easily.

Integrated management

In order to share experience with risk management, the integrated management leads risk management to an integral and vital part of the project management. So, any risks of the project will be taking care off in a short time. The integrated management adapts risk management methods and tools to a project's infrastructure and culture which provide the organizations reduce the impact of risk to other area of the organizations.

Continuous process

The continuous process is introduced to maintain constant vigilance to avoid the possibility of risks. This process identifies and manages risks routinely throughout all phases of the project's life cycle to ensure the project correction.

The organizations have to inform decisions by intentionally assessing on failures as well as the possibility and severity of the impacts of risk management. The informed decisions involve the evaluation of the trade-offs associated with all policies' options for risk mitigation in terms of their costs, benefits, and risks, and the evaluation of the impacts of current decisions on future options. This process of risk management embodies the identification, analysis, planning, tracking, controlling, and communication of risks.

Acquisition, development, and deployment programs continue to suffer from large cost overruns, the schedule delay, and the poor technical performance. Generally, this is a result of failing to deal appropriately with uncertainty in the acquisition and development of complex, software-intensive and software-dependent systems. The acquisition and development communities lack a systematic way of identifying, communicating, and resolving technical uncertainties.

This thesis focuses on the symptoms of cost overruns and the schedule delay rather than the root causes in the product acquisition and development. In fact, all areas in system development are potential sources of software risks as see in Figure 1, since it involves technologies, hardware, software, people, cost, and schedule [18].

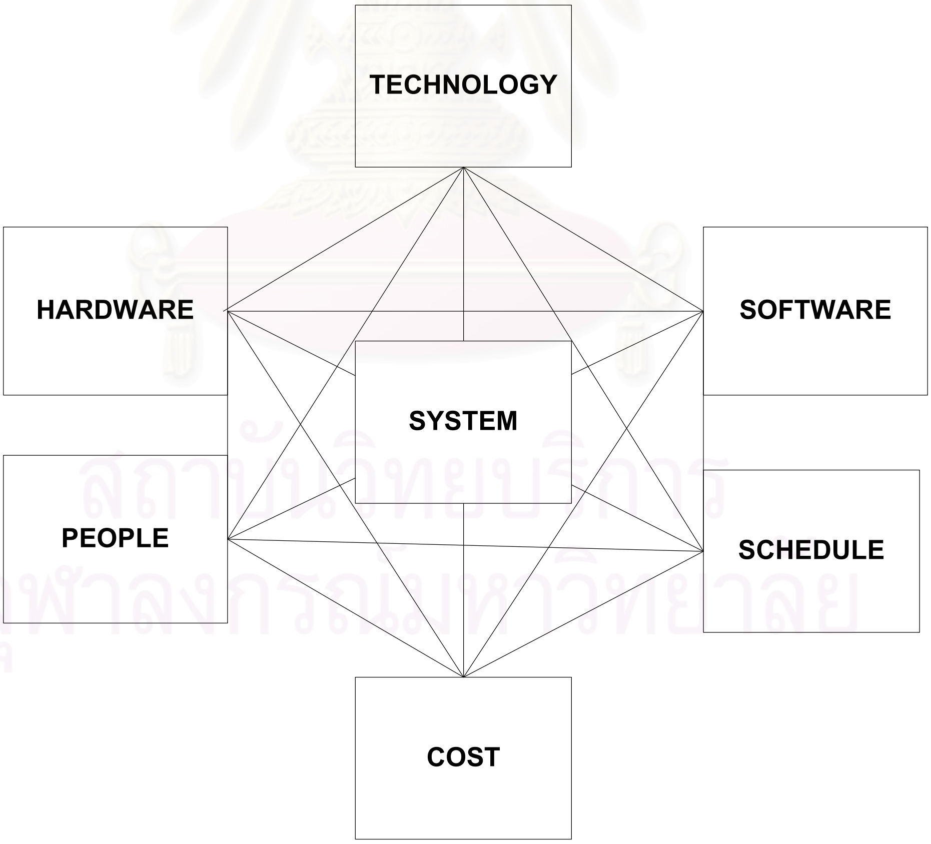


Figure 1: Risks within a System Context

In order to solve problems on cost overruns and schedule delay. The risk is commonly defined as a measurement of the probability and severity of adverse effects [24]. The software technical risk can be defined as a measurement of the probability and severity of adverse effects inherent in the development of software that does not meet its intended functions and performance requirements [21].

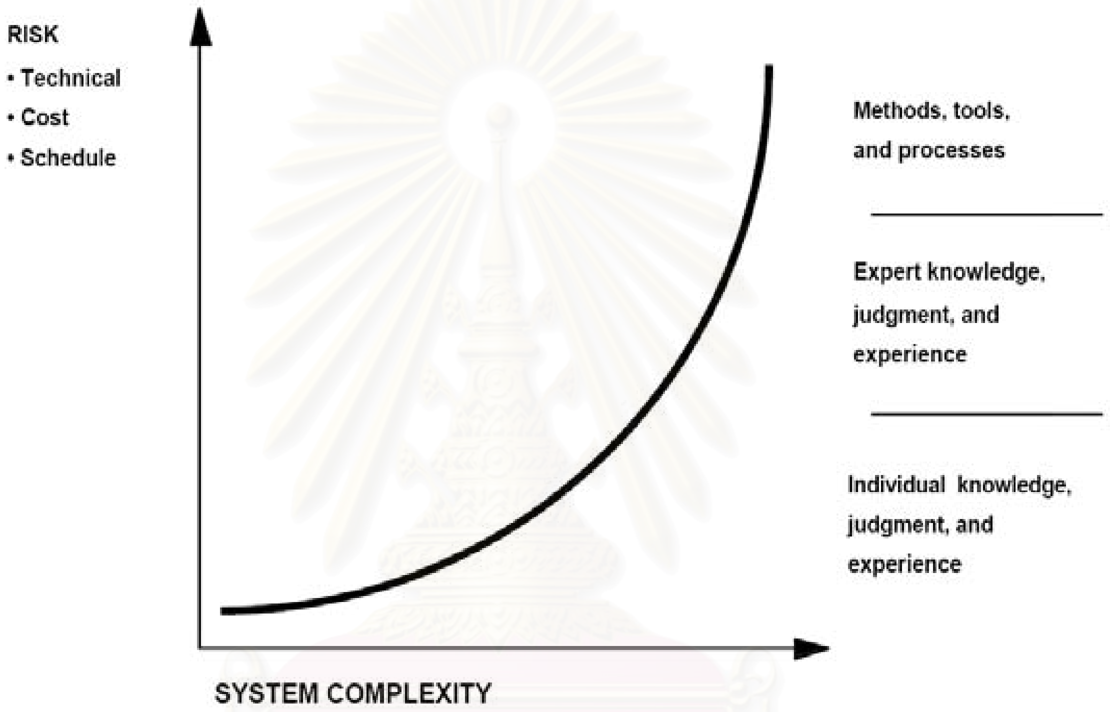

RISK
• Technical
• Cost
• Schedule
Methods, tools, and processes
Expert knowledge, judgment, and experience
Individual knowledge, judgment, and experience
SYSTEM COMPLEXITY

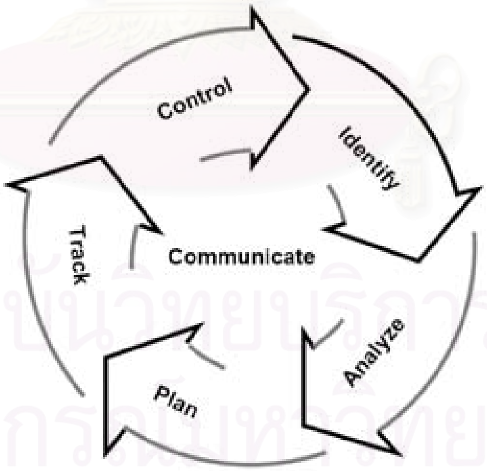
Figure 2: The Need to Manage Risk Increases with System Complexity [11]

The significant of risk management increases with the system complexities. Figure 2 demonstrates this concept by indicating that as the complexity of the system increases, both technical and non-technical (cost and schedule) risks increase. There are increasing demands for more systematic methods and tools to supplement individual knowledge, judgments, and experiences. These human traits are often sufficient to address less complex risks. It is worthwhile for managers to believe that they should manage risks in multifaceted dimensions. The fact of the matter is that they are merely managing costs and schedules along with isolated cases of technical risks. Many of the most serious issues encountered in the system acquisition are the results of risks that either remained unrecognized or ignored until they have already created serious consequences. This thesis also focuses on risk management according to its important because structured techniques can be effective in identifying risks, approaches, procedures, and techniques existing in risk mitigation.

Various researchers have shown that only few methods are managing risks in a systematic way. Unfortunately, these few methods approaches tend to be ad hoc, undocumented, and incomplete [22].

The key in responding to these problems is to improve the process for acquisition and development of software-intensive systems, in particular, identifying risk before problems occur. Furthermore, the researches also propose that the communication about risks should be performed in a positive aspect, nonthreatening. Finally, the managers must resolve technical risk in a cost-effective manner.

The three groups of methodologies (Software Risk Evaluation (SRE), Continuous Risk Management (CRM), and Team Risk Management (TRM)) are based on three basic constructs for risk management. All three constructs build on the seven risk management principles in previous.



A continuous set of activities to identify, confront, and resolve technical risk

Figure 3: Risk Management Paradigm [11]

Referring to Figure 3, the Risk Management Paradigm advocates a continuous set of activities to identify, confront, and resolve technical risk, depicts the different activities involved in management of risk associated with the software development.

The paradigm is represented by a circle to emphasize that the risk management procedure is continuous processes, while arrows show the logical flow of information between activities. Communication is placed in the center of the paradigm because it is both the conduit through which all information flow and the largest obstacle in risk management. Essentially, the paradigm is a framework for the software risk management.

A project can apply the above framework as if it is the best solution to protect the risk. A brief summary of each risk management paradigm activity is described below.

Identify

The identifying process is to draw out all risks before they become problems.

Analyze

The analysis step is the conversion of all risk data into decision-making information. Analysis provides the basis for the project manager to work on the right and the most critical risks.

Plan

The planning process responses in turning the risk information into decision and action procedures. Planning involves developing actions to address individual risks, prioritizing risk actions, and creating an integrated risk management plan. The plan for a specific risk can be drawn in many formats. For example:

- Planning mitigates the impact of the risk by developing a contingency plan.
- Planning avoids a risk from changing the product design or the development process.
- Planning studies the risk to acquire much information and better determine its characteristics to enable wiser decision-making.

The key to risk action planning is to consider the future consequences obtained from a decision pinned today.

Track

The tracking process consists of monitoring the status of risks and the actions taken to eliminate them. Appropriate risk metrics are identified and monitored to enable the evaluation of the status of risk mitigation plans.

Control

Risk control corrects deviations from planned risk actions. Once risk metrics and triggering events have been chosen, there is nothing unique about risk control. Risk control relies on project management processes to control risk action plans, corrects for variations from plans, responds to triggering events, and improves risk management processes.

Communicate

The risk communication lies at the center of the model to emphasize both its pervasiveness and its criticality. It is hard to gain the best risk management approach without effective communication. While communication facilitates interactions among elements of the model in Figure 3, there are higher level communications to be considered as well. In order to be analyzed and managed correctly, risks must be communicated between the appropriate organizational levels.

2.2.1 Software Risk Evaluation (SRE)

The SRE is a formal method for identifying, analyzing, communicating, and mitigating software technical risk. It is used by decision makers for evaluating and mitigating the technical risks associated with the project. The SRE is conducted at major milestones early and periodically in the acquisition life cycle. The primary SRE functions are detection, specification, assessment, and consolidation. Details of these functions are described below.

Primary SRE Functions

Four primary functions are identified in the SRE: detection, specification, assessment, and consolidation.

1. **Detection** is a function to identify software technical risks of a target project. This function ensures systematic and complete coverage of all potential technical risk areas. It also ensures efficiency and effectiveness through the use of appropriate tools and techniques. Risk detection in the SRE practice is performed by using the following methods:

- Questionnaire: questionnaire ensures complete coverage of all areas of potential software technical risks.
- Interview: selection of appropriate individuals and guidelines for the make up of the interviewing groups ensures coverage of all viewpoints, including software development and support functions, technicians, and managers.

2. **Risk specification** is the function of recording all aspects of the identified software technical risk, including its conditions, consequences, and sources. One representation of a software risk statement has several advantages. For instance, it serves as a simple, guiding structure for risk detection activities and for communicating risks coherently and with sufficient details. It captures components of the risks and simplifies the tasks of prioritizing, isolating the conditions within which the risk applies, and focusing the risk mitigation efforts to the sources of the risks. Additionally, the risk specification records sources of particular risks.

3. **Assessment** is a function that determines the magnitude of each software technical risks. By definition, a magnitude is the product of severity of impacts and the probability of an occurrence of the risks.

Risk statements are assessed at one of three levels of magnitudes: high, medium, or low. The level at which a particular risk is assessed depends on the separate assessments of its severity of impacts and its probability of occurrences.

4. **Consolidation** is the function of merging, combining, and abstracting risk data into concise chunks of decision-making information. This is necessary due to multiple risk detection activities that identify related risks from different sources.

Only that set of risk statements which meets the defined criterion are considered as candidates for consolidation. Candidate risk statements must meet one of the following criteria for consolidation:

- consolidate the information into fragmentation due to minor variations or different aspects of the same risk statement,
- consolidate the information in difference granularity. For example, a minor risk statement which is covered in the context of another risk statement of larger magnitude.

2.2.2 Continuous Risk Management (CRM)

The CRM is a principle-based practice for managing project risks and opportunities throughout the lifetime of the project. These principles provide effective approaches in managing risk regardless of the specific methods and tools. These principles are composed of three groups include core, sustaining, and defining.

Core Principle

An effective risk management requires regular attention to support the core principle of open communication. The professionals associated with a project are the most qualified to identify the risks. Open communication requires encouraging free flows of information between all project levels that every team member could experience this information easily.

Sustaining Principles

The sustaining principles focus on the conduction of the project risk management process. These are inward-directed, fundamental principles. If established early in the program and constantly take care of risks, the organizations should ensure that risk management flows on the way they have planed.

Integrated management

This principle helps to assure that risk management processes, paperwork, and discipline are consistent with established project culture and practices. Risk management is simply an area of emphasizing in good project management. Therefore, risk management tasks should be integrated into a well-established project routine. Integrated management tasks are:

- Making risk management an integral and vital part of project management that provides a quick response to the project's risk.
- Adapting risk management methods and tools to a project's infrastructure and culture; so, the impacts of risks to other areas of the organizations are reduced.

Teamwork

No individual person can anticipate all the risks that face a project. Risk management requires that project members find, analyze, and work on potential risks as a team. Teamwork requires working cooperatively to achieve a common goal to reduce opportunity of risk in human dimension. Furthermore, it needs to exchange talent, skills, and knowledge among team members in order to improve performance of the rest of the team members.

Continuous process

The processes must be part of daily, weekly, monthly, and quarterly project management. The premise that risk management takes place only during the risk management process is obviously unfamiliar to suitable management. Continuous process requires maintaining constant vigilance to avoid the possibility of risk happening, including identifying and managing risks routinely throughout all phases of the project's life cycle to ensure the project correction.

Defining Principles

The defining principles focus on the organization's process to identify and address uncertainty risks. These principles are outward directed and concerned with care. This encourages the development of shared mental models that clarify occurrences of risk management.

Forward-looking view

This principle develops the ability to look ahead, beyond risk crisis and impacts of current decisions on future options. Risk management staff is also concerned with defining the future purpose, so that all risk mitigation efforts of project's staff are complementary.

Forward-looking view introduces project members to think towards tomorrow, identify uncertainties, and anticipate potential outcomes of the project which can predict occurrences of risks. Additionally, it manages project resources and activities while anticipating uncertainties that can decrease possibilities of risks.

Global perspective

This principle requires that the project staff replace their narrow views and interests with the benefit of overall project. The project staff should develop and share common viewpoints at a global level, and be able to jointly address and mitigate specific risks.

Global perspective views software development within a context of a larger system-level definition, design, and development to ensure system correctness. In addition, the global perspective also recognizes both the potential value of opportunity and the potential impacts of adverse effects, such as cost overruns, time delay, or failures to meet product specifications. This provides organizations in preventing risks that might occur in the system.

Shared product vision

This principle focuses on the development of a common understanding of project's objectives. Shared product vision makes it much easier to reach a common understanding of risks' impacts to the timeliness, costs, or features of the final result.

Sharing product vision is based on a common purpose of software purchasing to make the organizations confidence on their objectives. This is also based on collective commitments of decision making that guarantee the discussion results are correct. This method reduces the chance of system failure that the organizations are missing their goal.

2.2.3 Team Risk Management (TRM)

TRM extends risk management with team-oriented activities involving the user and vendor, where both user and vendor apply the methodologies together. TRM establishes an environment built on a set of processes, methods, and tools that enables the user and vendor to work cooperatively, continuously managing risks throughout the life cycle of a software-dependent development program. It is built on the philosophy of cooperative teams. TRM further extends the Risk Management paradigm by adding two functions include initiate and team [24]. Each risk goes through these functions sequentially, but the activity occurs continuously, concurrently, and iteratively throughout the project life cycle. The TRM provides an effective instrument with the concepts, functions, processes, methods, and products of TRM. It accomplishes this through a description of the overall methodology, a road map for applying it within a project, and detailed descriptions of the processes and methods used to implement the functions of TRM.

Initiate

Initiate recognizes the requirement and commit to create the team culture. Either user or vendor may initiate team activity, but both must commit to sustain the teams.

Team

Formalize the user and vendor team and merge the viewpoint to form a shared product vision. Systematic methods periodically and jointly applied establish a shared understanding of the project risks and their relative importance. Establish joint information base on risks, priorities, metrics, and action plans.

2.3 Software Process Model

Since 1960s, many descriptions of the classic SDLC have appeared originated the formulation of the SDLC using waterfall chart as shows in Figure 4[19]. The chart summarizes in a single display to present the difficult of developing a large software system because it involves complex engineering tasks that may require iteration and rework before completion. These charts are often employed during introductory presentations, demonstration for people who are in charge of making decision, and software development team members when implementing a new software.

These SDLC models usually include the following activities [47]:

System Initiation/Planning: this activity defines user requirements, work effort, and project timeline.

Requirement Analysis and Specification: this activity identifies the problems of the new software system which are supposed to be solved. The operational capabilities, desired performance characteristics, and the resource infrastructure are required to support the system operation and maintenance.

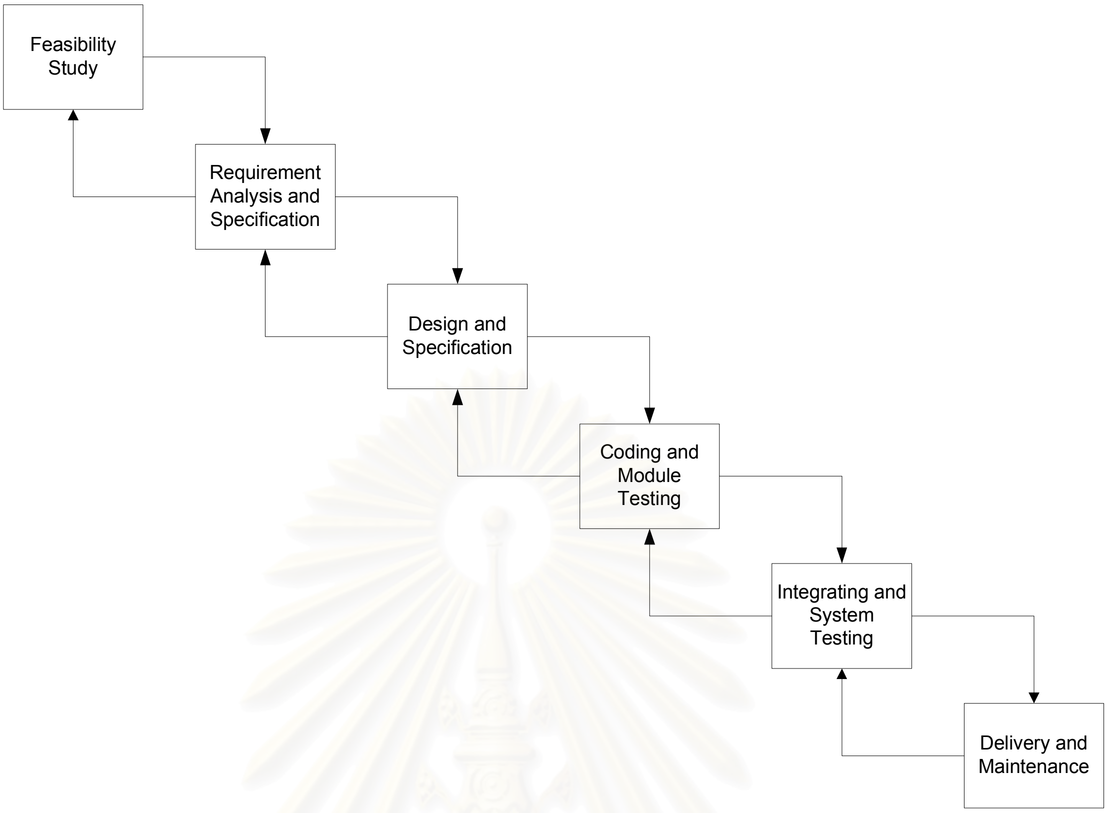

Feasibility Study
Requirement Analysis and Specification
Design and Specification
Coding and Module Testing
Integrating and System Testing
Delivery and Maintenance

Figure 4: The Waterfall Model of Software Development

Functional Specification or Prototyping: this activity identifies and potentially formalizes the objects of computation, attributes and relationships, the operations that transform these objects, constraints that restrict system behavior, and so forth.

Partition and Selection: this method provides requirements and functional specifications, this process divide the system into manageable pieces that denote logical subsystems which can determine new, existing, or reusable software systems correspond to the needed pieces.

Architectural Design and Configuration Specification: this method defines the interconnections and resource interfaces between system and subsystems, components, and modules, in the suitable ways for detailed design and overall configuration management.

Detailed Component Design Specification: this activity defines the procedural methods through the data resources within modules of a component that are transformed from required inputs into provided outputs.

Component Implementation and Debugging: this process codifies the preceding specifications into operational source code implementations and validates their basic operations.

Software Integration and Testing: this method affirms and sustains the overall integrity of the software system architectural configurations through verifying the consistency and completeness of implemented modules. Additionally, this process verifies the resource interfaces and interconnections against their specifications, and validates performance of the system and subsystems against their requirements.

Documentation Revision and System Delivery: the purposes of this method are packaging and rationalizing recorded system development descriptions into systematic documents and user guides; all are in the suitable forms for dissemination and system support.

Deployment and Installation: the objectives of this activity are providing directions for installing the delivered software into the local computing environment, configuring operating systems parameters and user access privileges, including running diagnostic test cases to assure the usability of basic system operations.

Training and Use: the aim of this process is to provide users with instructional aids and guidelines for understanding system capabilities and limits, in order to use the system effectively.

Software Maintenance: the purpose of this method is sustaining useful operations of the system in its host/target environment by providing requested functional enhancements, repairs, performance improvements, and conversions.

A SDLC model is either a descriptive or prescriptive characterization of developed software. A descriptive model describes the history and development of a particular software system. Descriptive models may be used as the basis for understanding and improving software development processes or for building empirically grounded prescriptive models. A prescriptive model prescribes the direction to develop a newly software. Prescriptive models are used as guidelines or frameworks to organize and structure software development activities [48]. Typically, it is easier and much common to articulate a prescriptive life cycle model for the direction to develop software systems. This is possible since most models are intuitive or well reasoned. This means that many distinctive details that describe the structure of a software system can be ignored, generalized, or deferred for later consideration. Thus they should raise concern for relative validity and robustness of such life cycle models when developing different kinds of application systems, in different kinds of development settings, or using different programming languages with differentially skilled staff, etc. However, prescriptive models are also used to package the development tasks and techniques for using a given set of software engineering tools or environment during a development project.

On the other hand, descriptive life cycle models characterize the specific partial development of software systems. Thus, they are less common and much difficult to articulate for obvious reasons. For example, they must observe or collect data throughout the life cycle of a software system, a period of elapsed time often measured in years. Also, descriptive models are specific to the systems which only generalization through systematic comparative analysis. Therefore, this suggests the prescriptive SDLC models will dominate attention until a sufficient base of observational data is available to articulate empirically grounded descriptive life cycle models.

These two characterizations suggest that there are a variety of purposes for articulating SDLC models. These characterizations serve as follow:

guideline to organize, plan, budget, schedule, train, and manage software project work over organizational time, space, and computing environments,

prescriptive outlines for producing documents delivered to clients,

basis for determining proper software engineering tools and methodologies will be most appropriate to support different life cycle activities,

framework for analyzing or estimating patterns of resource allocation and consumption during the SDLC,

basis for conducting empirical studies to determine factors that affect software productivity, cost, and overall quality.

Software process models often represent a networked sequence of activities, objects, transformations, and events that embody strategies for accomplishing software evolution [49]. Such models can be used to develop more precise and formalized descriptions of SDLC activities. Their power emerges from their utilization of a sufficiently rich notation, syntax, or semantics, often suitable for computational processing.

2.4 Business Process

A business process is defined as a structured collection of tasks animated by actors to transform resources into services. A process can be decomposed into a set of sub-processes collaborating to build overall services. Many other definitions of a business process can be found in the literature [25][26][27][28].

Services from the business process can be counted as distinct outcomes to ensure stakeholders for correctness of their decisions. These are the entities that directly or indirectly benefits from the services of the process. The stakeholders assess the process outputs against their success criteria that defined their expected value from the services. The stakeholders can either be the final consumers of the services or receive the service directly.

The process that delivers services to stakeholders is called a service provider [44]. This process is separated from its environment by the process boundary. It delivered through a part of its process boundary called a service interface and received by stakeholders through a part of its process boundary called a use interface [45].

The nodes in the diagram of Figure 5 are the set of states of the constituent process elements: tasks, processed items, and actors. When processes are decomposed into sub processes, the total state of the super process is the set of the external states of its sub processes.

The service provider exhibits only a part of total state at the service interface. The exhibited part of the service provider’s total state is an external state while the remaining part is internal states. The provider is delivered as a sequence of its external state exhibited at the service interfaces and perceived by stakeholders at their use interfaces.

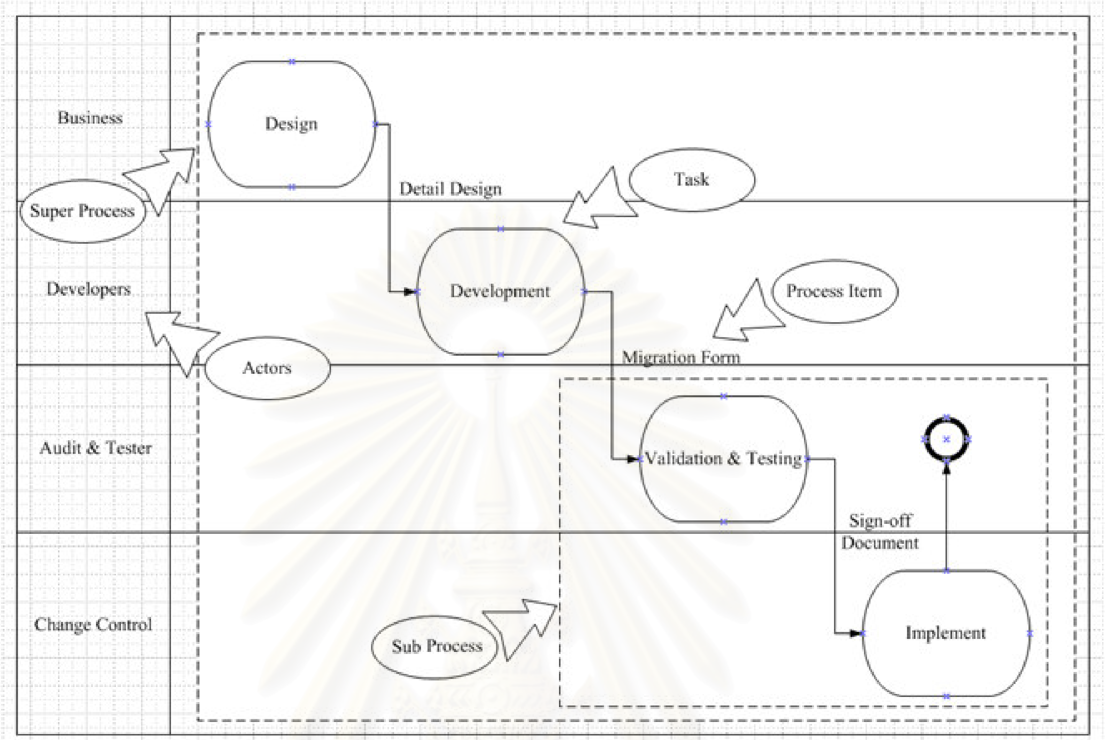


Figure 5: Business Process Diagram

Services are correct if they meet all success criteria of all stakeholders. Service utilities are subjective to stakeholders' estimation of the value delivered by the services to stakeholders.

2.5 Requirement Analysis

Requirements [29][30][31] are descriptions of a system behavior or descriptions of system properties or attributes. It can alternatively be statements of an expected application.

The software requirement analysis process covers the complex tasks of eliciting and documenting requirements of all users, modeling and analyzing requirements and documenting as a basis for system design.

Software requirement analysis and documentation processes are critical to the success of a software project [32]. The requirement engineering method is an emerging field which deals with the systematic handling of requirements.

Software requirement analysis consists of 6 steps include[33]:

2.5.1. Fix system boundaries

This initial step is the integration of the purchasing software with the business processes, fitting it into the larger picture and defining scope and limitations of the software.

2.5.2. Identify customers

In the early stage, researches and development had been focused on identifying users or customers of applications. Referring to the stakeholders, these indicate the group or groups of people who will be directly or indirectly impacted by purchasing software. The requirement elicitation process should focus on the wish list of this defined group to obtain a valid requirements list.

2.5.3. Requirement elicitation

Information is gathered from multiple stakeholders. The requirement analysts are selected from these groups in order to extract their requirements. Considering multiple stakeholders involved, a list of requirements could run into pages. The depth of each requirement is based on the number and size of user groups, the degree of complexity of business processes, and sizes of applications.

The problems faced during the requirement elicitation period are: ambiguous understanding of requirements, inconsistency among requirements obtained from multiple users, insufficient inputs from stakeholders, conflicting interests of stakeholders, change in requirements after project has begun.

A requirements analyst has to interact closely with multiple work-groups, often with conflicting goals, to obtain a requirement list. Strong communication and people skills along with the programming knowledge are prerequisite for expert requirement analyst.

Traditional methods of requirement elicitation include stakeholder interviewing and focus group studies [34]. Other methods like flowcharting of business processes, and the use of existing documentation like user manuals, organizational charts, process models and systems or process specifications, on-site analysis, interviews with end-users, market researches and competitor analysis were also used extensively in the requirement elicitation process.

However the current research in the software requirement analysis process has thrown up the modern tools that are better equipped to handle the complex [35] and multilayered process of requirement elicitation. Some of the current requirement elicitation tools are: prototype, Use cases, data flow diagrams (DFD), Transition process (TP), and User interfaces (UI).

2.5.4. Requirement analysis Process

Once all stakeholder requirements have been gathered, a structured analysis of these can be performed after modeling these requirements. Some of the applied software requirement analysis techniques are requirement animation, automated reasoning, knowledge-based critiquing, consistency checking, analogical and case-based reasoning.

2.5.5. Requirement Specification

Requirement elicitation, requirement modeling, and requirement analysis should be clearly documented. A written requirement document is significant so that it must be drawn out from all stakeholders, including clients, user groups, developing and testing teams.

Requirement specification is vital and serves as:

base for validating the stated requirements and resolving stakeholders' conflicts, contract between clients and the development team

basis for system design for the development team

benchmark for project managers to plan the project development lifecycle and goals

source of formulating test plans for testing teams

resource for requirement management and requirement tracing

basis for evolving requirements over the project life cycle

Software requirement specification involves scoping the requirements so that it meets customers' visions. It is the result of collaboration between end-users, and technical/systems analysts.

The software requirement specification is a document that lists out stakeholders' needs and communicates these to the technical community that will design and build the system. The challenge of a well-written requirement specification is the clear communication to both groups and all subgroups [42].

To obtain a well-written requirement specification document, it must be documented separately as

User Requirements – user requirements have been written in clear, precise language with plain text and use cases, for benefits of customers and end-users.

System Requirements – system requirements have been expressed as programming or mathematical models, addressing for the application development and testing teams.

Requirement specification serves as a starting point for software, hardware and database designs. It describes functional and non-functional specifications of the system, performance of the system, and the operational and user-interface constraints that will govern system development processes.

2.5.6. Requirements Management

Requirement management is the comprehensive process that includes all aspects of software requirement analysis and additionally ensures verification, validation, and traceability of requirements. Effective requirement management practices guarantee that all system requirements are stated unambiguously, omissions and errors are corrected. Additionally, it evolves specifications that can be incorporated later in the project lifecycle.

2.6 Product Prototype Analysis

Product prototype analysis [38][39][40] involves the production of functionally useful and trustworthy systems through experimentation with evolving systems. Generally, this experiment conducted with many users involvement in the evaluation of the prototype [43]. The results of this analysis can determine common features that are the greatest number of individuals. Importantly, the product prototype analysis shows concepts that are unique to a particular group of software [46]. For these reasons, product prototype analysis has recently become widely accepted as a method to systematically examine and compare subjective or concepts, and is the model used for this study.

2.7 Proof of Concept

Proof of concept (POC) [36][37] requires an agreement set of proofs or tests that defines its success criteria. Proof requirements are generally agreed and documented after the scope of the document is completed and before defining solutions. This document, called as a proof requirement document (PRD), defines the proof context and identifies individual components of proof requirements. Consequently, each proof includes requirements, proof methods, acceptance criteria, and success measurement indicators.

Proof requirements are related to objectives and goals established in the PRD. This document can be separate into two parts as follow:

Proof Requirements: The proof requirements are used to drive the POC solution design, and manage project's scopes during subsequent phases of the POC. It is used as a reference and a driver for the proof presentation and documentation; and it can also be used to record results against each required proof.

Objectives and Scope: The POC objectives, scope, and overall solution visions are defined in the PRD. The PRD defines how the solution visions will be confirmed by the POC.

The previous studies conclude that the POC process normally occurs after software is selected [41]. It is a major undertaking and should not be used as a tool to compare one solution over another. Furthermore, as the POC is orientated to the chosen solution and implementation methodology, the content or process undertaken cannot be readily transferred to other software solutions. Using only the POC could be used to confirm the preferred vendor's status rather than establishing it.

There are a number of advantages to the POC for organizations including:

Organizations could meet expected synchronization with vendors. In addition, organizations could extract and identify software functional gaps or overselling. In each evaluation phase, evaluated members have to test and track software based on organizations' requirements.

The POC improved the correctness of the scope of the document which provides a better understanding of organization requirements to complete the implementation. The POC process allows the organizations to evaluate the software and implementation vendors.

However there are limitations of the POC that should be noted as follows:

depending on the commercial agreements, the POC can be placed within the sales cycle; thus, software vendor is restricted for a full disclosure. Further, the documentation produced within the POC may have marketing content which does not add value to the project.

A POC should be completed as a part of the selection process when the risk of project failure is comparatively high. Risks can be measured by two key variables: complexities of requirements, and levels of expertise within the selection team members.

2.8 Software Demonstration

Software demonstration [30] displays software behavior, features and functions. This process can identify software weak points. This process could help organizations in making decision. It shows software feasibilities, complexities and user friendly. The organizations could experience that the purchasing software has ability to response their expectations.

2.9 Software Presentation

With similar to software demonstration, software presentation process [30] shows software advantages, features, and functions, except that organizations cannot run the purchasing software. It also presents software interfaces, software supporting languages and minimum requirements of the system.

2.10 Learning software behavior and vendor service from other organization

The risk management process on learning software behavior and vendor services from other organizations can recognize vendors' reputation and reliable services in a shortest time. Thus, vendors with bad reputation can be eliminated. Consequently, organizations can avoid disasters during implementing the purchased software. Furthermore, organizations could obtain important information of purchasing software from others in term of services, functionalities, compatibilities, usability and flexibilities.

Although, the standard risk management procedures have abilities to eliminate risks on purchasing the worthless software, unfortunately each procedure also has disadvantage features. Therefore, applying a standard procedure does not guarantee that the delivered software will be completed as needed. Thus, in order to avoid the software failure, all business organizations have to determine their own standard risk management process that ensures their needs to be served. Therefore, this research will focus on the risk management process that the business organizations can really applied to support their satisfactions in software purchasing.

This thesis proposed the combination on risk management process with proof of concept, product prototype analysis. The detail and result of proposed method is presented in the next chapter.

CHAPTER III

ORGANIZATIONAL SURVEYS METHOD

In the business section, software failures can cause dramatically damage to the entire organizations. Therefore, selecting software to be implemented is an important issue that every manager must be aware. Since most business organizations have different business profiles and objectives, the software purchasing process for each organization is usually different from each other. So, this research, the study of software purchasing has been performed using depth interviewing, including distributing the questionnaires to various organizations with different sizes and different business objectives. Figure 6 shows the elicitation methods of this thesis.

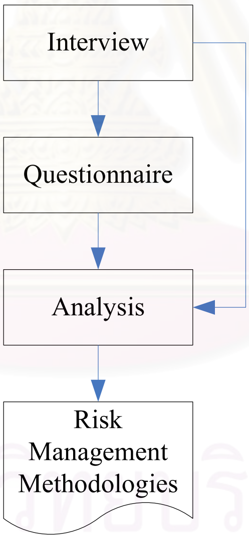


Figure 6: Organizational Surveys Method

Referring to Figure 6, the first elicitation method for acquiring information is interviewing which inquiring business profiles, business sizes and business objectives, discussing their selecting risk management process, succession and problems. The information that is extracted from the interviewing process is exploited to build the questionnaire. Additionally, the questionnaire was distributed to various business sizes and business objectives with IT-related based in order to capture the users' satisfactions of risk management methods for risk analysis. The next step is the analysis step which analyzes information from interviewing process and questionnaire. This step separately examines the information by the business's objectives based on types of organizations: application-based organizations, or computer-based organizations. The outcomes of the analysis process are risk management methodologies that present the user satisfaction rate of applying each selecting risk management process. These results are used to indicate the preference in each risk management methodology.

Sampling data was collected form IT-related based business on both computer-based and application-based organizations, including all business classifications with different business sizes, such as hospitals, airlines, communications, information technologies, financial and banking, etc.

The business size can be classified into 3 groups that are large, medium and small defined by Thai's Ministry of Industry. Organizations that have investment budget less than 5 million baht are defined to be small. Organizations that have investment budget during 5-200 million baht can be defined to be medium. Organizations that have investment budget greater than 200 million baht are classified as large organizations.

In this research, the study of software purchasing has been performed using depth interviewing, and distributing the questionnaire to various organizations with different sizes and different business objectives. The details of these processes are described in the following sections.

3.1 Interviewing Process

The interviewing process has performed in 8 companies. The interviewing surveys method is performed by walking in to organizations. Every interviewee is in the rank of manager or president of the company who authorizes in software purchasing process of the organization; the interviewing time of each place is about 1 hour. Moreover, the sizes of these companies can be categorized in 3 large companies, 2 medium companies and 3 small companies.

The objectives of this process are to extract organizations' expectations, and interview their profiles as follow:

Part 1 Inquiring business profile, business size, and business objectives to categorize each organization

The objectives for finding these information are to categorize each business objective into organizational-based: application-based and computer-based, and classified each organization into groups based on business sizes.

Part 2 Interviewing organizations about the current processes that used in purchasing software, and the reason on applied that processes

The purposes are acquiring their software purchasing process for investigating performance, and inspiration of each software purchasing process.

Part 3 Discussion on the succession of current software purchasing processes, and problems of them

The aims of this process are exploring the successes of each software purchasing process, effects of software failure, including actions to solve the failure problems.

The results from this interviewing will be used to create the questionnaire for further purchasing techniques from the business companies in details. Moreover, these outcomes will also used in the analysis process to indicate the best effective method in software purchasing procedures that organizations should apply.

3.2 Questionnaire for Elicitation Process

In order to obtain the real risk management criteria in the business area for software purchasing, another elicitation method to capture decision factors of business organizations is the use of questionnaire. The questionnaires are distributed to 110 companies; all can be classified as 24 large, 71 medium, and 15 small all the business sizes are classified the same as the companies in the interviewing process.

The objectives of these methods are to collect information about software purchasing process of organizations. The content of the questionnaire are organized as follow:

Part 1 Querying organizational profile, business size, and business objectives to categorize each organization

The purposes of querying are the same as the interviewing process mentioned above.

Part 2 Questions about the organizations' purchasing software processes, and reasons to apply the processes

The questions in this part are similar to the interviewing process, except that the objective of this distribution is to confirm the procedures obtained from the interviewing information. Moreover, some questions are added to clarify the reasons of purchasing or developing software for the organizations.

Part 3 Inquiring for the satisfaction of their processes, problems and solutions of their problems

According to the interviewing process, the rating of customers' satisfaction must be rated. This part allows managers to rate the purchasing software obtained from their selection processes. The outcome of this rate can indicate success or failure of the purchasing software based on the managers' expectations.

Generally, the satisfaction rate is low if and only if software cannot serve the client's needs. Thus, some functions cannot be served as planned, operational problems occur as a consequence. This questionnaire will also capture all problems under this situation include solutions to solve such problem.

Part 4 Further recommendation from the organizations

In order to acquire others useful information for analyze, the recommendation is used for gather other important information that organizations would like to share.

3.3 Questionnaire for Evaluation Process

This evaluation process performs based on elicitation process. This process tries to capture user expectation of the results of using combinations of various risk management procedures, the POC, product prototype analysis, and learning software behaviors and vendors' services from other organizations. Furthermore, this evaluation process lets organization applied the recommended process for 6 months.

The objectives of these processes are to acquire user satisfaction level of the expected procedure after applying the expected procedure. The content of the questionnaire are organized as follow:

Part 1 Inquiring for the satisfaction of the expected method and problems

According to the elicitation process, there is another expected process to be discovered, is combinations of various risk management procedures. This part allows managers to experience the expected method for awhile. The outcome of this process can indicate success or failure of the suggested method based on the managers' satisfaction. Additionally, this questionnaire will also capture the problems occur by using expected procedure.

Part 2 Further suggestion of the recommended method

In order to improve the purchasing software process, the other suggestion of using the expected method has to be addressed.

CHAPTER IV

SURVEYS RESULTS

After the organizational surveys, the summarizations of the interviewing and surveys results are presented in this Chapter. Firstly, this thesis explains software selection process that used in Thai's organizations. Secondly, this thesis discusses in risk factors, risk dimensions, and risk technique. Thirdly, it describes consideration of risk factors for software purchasing of organizations. Next section, it introduces risk management processes that applied in software purchasing processes. The rest of this Chapter are the comparisons of user satisfaction rate against risk management methods and propose the appropriate risk management procedure.

4.1 Software Selection Process

Referring to the interviewing and questionnaire described in Chapter 3, the software selection process can be summarized as follow.

- Read advertisement on computer magazines, open vendor auction, or searched the internet on software ranking websites.
- Announce organization requirements and let software's vendor to take auction.
- Provide organization system overviewed to selected vendors. Organizations can analyze possibility to implement their software.
- Set up software purchasing project team members. Team members include system analysts, developers, business analysts, and users.
- Invited selected vendors to present their software features, implemented processes and plans.
- Applied organizational risk management process for decision making on software purchasing. Thai's organizations typically use at least two of software selection processes above to obtain the purchased software.

The results from the elicitation process show that there are different risk management processes among different sizes of organizations, with irrelevant to the business profiles.

The following section discusses on risk factors, risk dimensions and risk techniques.

4.2 Risk Factors, Dimensions and Techniques

There are many different components of risk management methods, as shown in Table 1 which provides some examples of the risk factors against risk dimensions and techniques. Different risk management techniques are applicable not only to different risk components but to each risk dimension. While there are some commonalities among these dimensions by focusing on their keys which can better understand the management of dimensions, thus, organizations could manage their risk dimensions with their appropriate risk techniques.

The major dimensions of risk management in software purchasing are functions, financial, maintenance cost, meet requirement, versatile, and compatibility. Requirement analysis, proof of concept, product prototype analysis, software demonstration, software presentation, and learning software behaviors and vendors' services from other organizations are risk management procedures used by Thai's organizations which struggled to minimize loss in their application of this new purchasing technology. The risk dimensions have been most widely addressed in the literature; numerous procedures have been developed to manage these dimensions. However, these risk management strategies must constantly be refined because software technologies are complex, continually changing, and applied to unique problems.

Table 1: Risk Factors, Risk Dimensions, and Risk Techniques

Risk Factors	**Risk Dimension**	**Risk Technique**
Functions	Misunderstood requirement	Requirement analysis
Financial	Cannot accurately estimate benefits	Full cost analysis; Competitive benefit assessment
Maintenance Cost	Cannot accurately estimate costs	Estimation models; Modularization; Project management tools; Software reuse
Meet Requirement	Inadequate plans or procedures	Process control plan
Versatile	Inability to do more complex function	Additional subsystem interface
Compatibility	Incompatible with legacy system	Use standard software language based and standard
Maintainability	Poor design, code, maintenance procedure	Tracing tools; Automated restructure; Debuggers, coding Standard
Reliability	Inadequate measures	Reliability models
Safety	Inadequate assessment tools	Fault tolerant design, system safety analysis
Performance	Inadequate simulation tools	Simulation modeling; Benchmarking
Personal	Personal lacks necessary technical skill	Staffing and Training including new recruit

The risk factors for software purchasing of each company's size will discuss in next section.

4.3 Consideration Factors for Software Purchasing of Each Company's Size

There are mainly two influence types that major concerned: business influencers, and technical influencers. It is obvious that the business size effects on the software purchasing process. By the business size, small business organizations usually avoid using complicated risk management process in case of reducing time and tasks, while the larger business sizes concentrate on complicated risk management processes because it could produces better results than ease tasks.

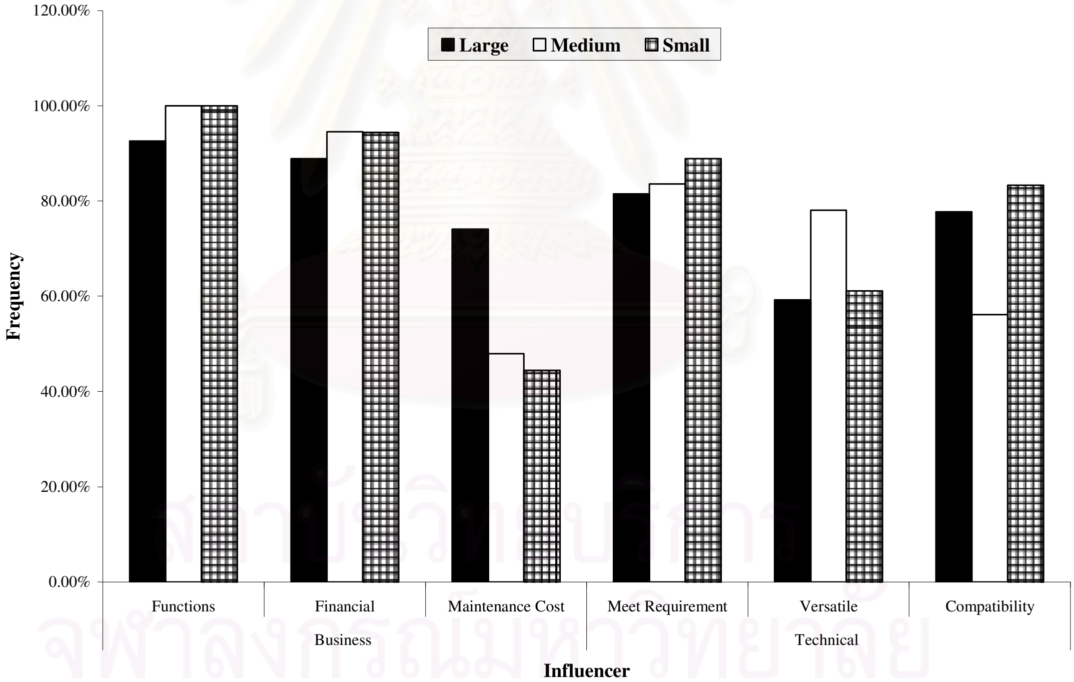


Figure 7: Consideration Factors for Software Purchasing of Each Company's Size

Figure 7 shows the consideration factors for software purchasing of each company's size, it is obvious that 100% of small and medium size organizations give priorities to functions to be performed while large size organizations are also highly focusing in functions which mean that the main objective of every organization is the functions of the purchasing software. In addition, the second significant factor is financial, denoted that this factor is also important in every business size as it presents more than 90% of every organization size. Similar to the financial factor, requirement satisfaction presents the third importance which shows more than 80% of all sizes.

However, under other factors, Figure 7 shows that the large size organizations are serious of the maintenance cost more than other business sizes due to the value that has to spend on purchasing software and the complicated of system which also increase for maintenance cost. On the other hand, small and medium business sizes have less concern in this factor because the value of software and the complicate of their system much less than the large size companies. About 80% of medium size organizations are focusing in versatilities which their various business strategies of medium size companies is not too much difference like large size companies and small business size mostly have only one business strategy which do not necessary to worry about this factor. Considering the compatibility factor, the survey result shows that about 60% of medium size organizations take account in this factor; this is because the size of the organization is less complicated than small and large organizations which can be counted as 80%. The reasons for this different is that almost small and large size organizations usually have different operating system platforms which have to apply purchasing software into various types of platforms.

Focusing on business objectives, the financial and banking organizations, mainly, concerns in functions, financial, requirement satisfaction and compatibility while organizations in information technology concerns in functions, financial, and requirement satisfaction. Furthermore, most computer-based organizations are focusing in functions, requirement satisfaction, versatility, and requirement satisfaction of purchased software. On the other hand, most application-based organizations are concerning in functions, financial, requirement satisfaction, and compatibility.

As the consequence of the results mentioned in the previous section, the important factors that every company must determine before the risk management starts, are financial, functions, maintenance, compatibilities, and requirements fulfillment.

4.4 Risk Management Process for Software Purchasing

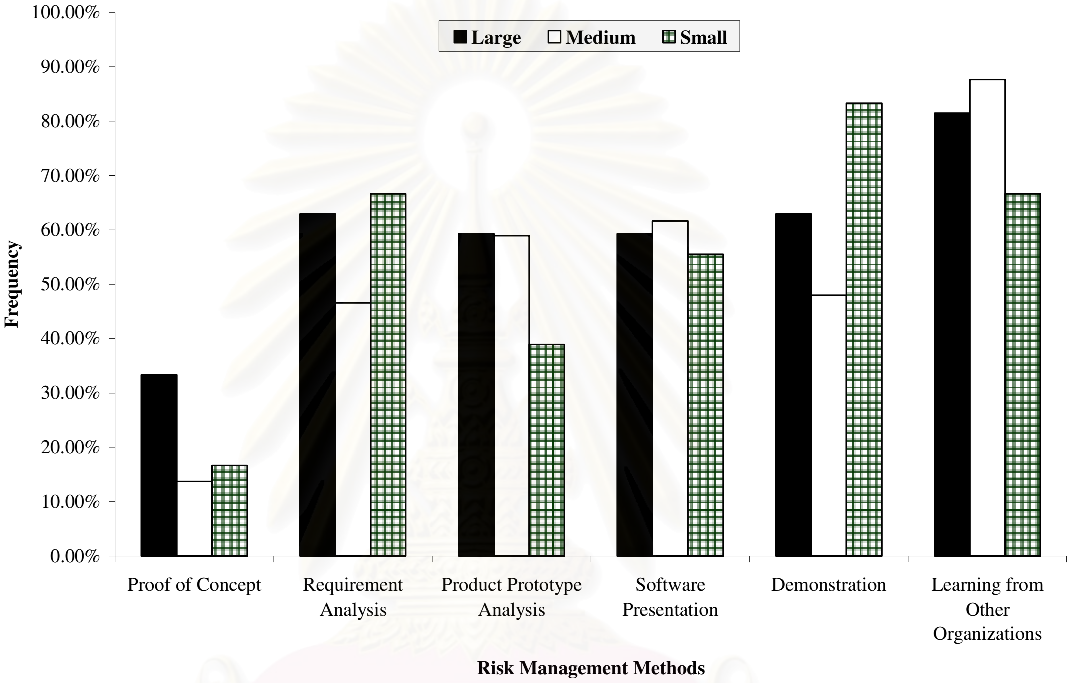


Figure 8: Risk Management Process for Software Purchasing

Figure 8 shows the frequency of the risk management process for software purchasing that applied in sampling Thai's organizations which separate by business sizes. It is obvious that the most widely used risk management methods of Thai's companies is learning software behaviors and vendors' services from other organizations because it is the easiest way to obtain software behaviors and vendors' services without taking much effort. While POC presents the lowest used from the surveys. The standard risk management procedure which is requirement analysis is the moderate use on every business size.

Focusing on each business size, the most favorite risk management method for small size organizations is software demonstration which shows 85% of the surveyed organizations that applied this method. According to previous figure, Figure 7, the small size organizations usually apply functions and compatibility factors in purchasing software; thus, software demonstration has ability to handle these factors. While more than 80% of medium size organizations and large size organizations are likely to apply learning software behavior and vendor's services. This process is highly used for the reason that the obtained information from other organizations are mostly reliable due to the organizations' reputation. The organizations that applied the purchasing software have the potential to expand their business strategies and market share. Based on the POC rate, its shows the minimum use rate in every business size because this process requires too much effort, and business and veteran staffs in order to proof and test the success criteria of the purchasing software. In performing POC, organizations have to set up many team members to handle the whole project. As a result, they have to recruit new staffs to handle these processes which increase the overall budget.

The User Satisfaction Percentage = $\frac{\sum_{i=1}^{5}(N_i * i)}{M * 5} * 100$

N = Total Number of Organizations that present in each satisfaction level.
i = Satisfaction Level value = 1,2,3,4 and 5; the greater means the higher satisfaction.
M = Total Number of Organizations use each risk management process.

The above formula is used to calculate a user satisfaction rate of each risk management method within the rest of this thesis.

4.5 User Satisfaction Rate on Each Risk Management Process

Figure 9 shows the user's appreciation rate of computer-based organizations comparing with users' satisfactions factors using every focusing risk management methods. The POC demonstrates the highest user acceptance level, even though it is not widely used in many organizations because it has to perform many complicated tasks and take a large number of resources. In contrast, the software presentation method shows the lowest user acceptance rate while the requirement analysis method shows the moderate level of user satisfaction.

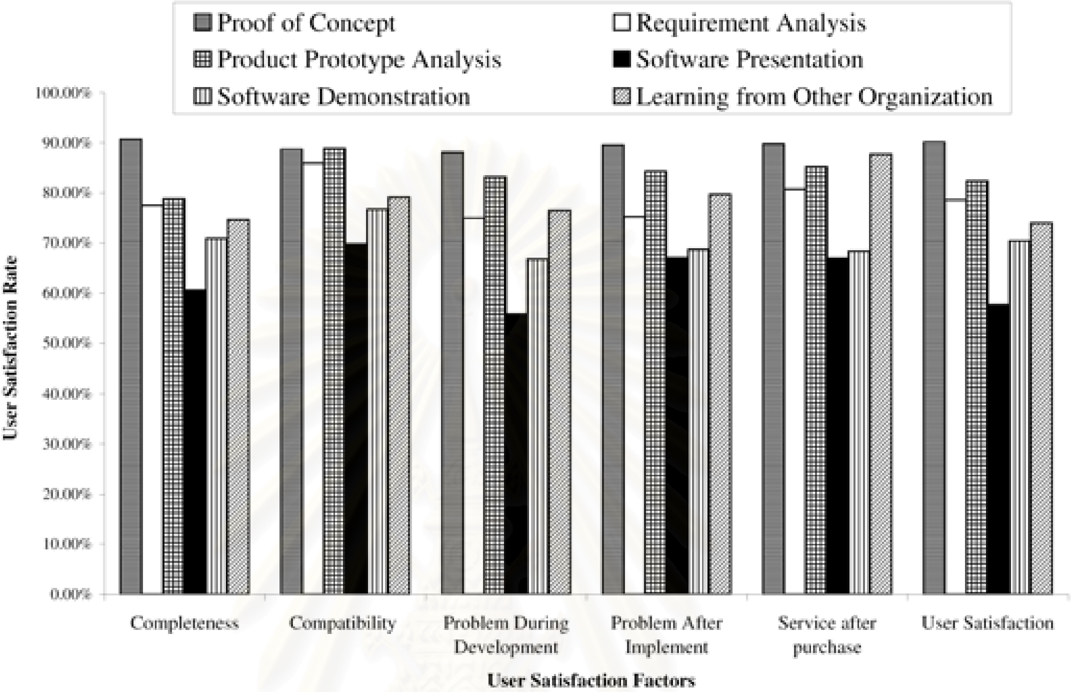


Figure 9: User Satisfaction Rate on Risk Management Process with Computer-Based Organizations

As of Figure 9, POC presents almost 90% user satisfaction in every user perception factors. According to the complicated tasks to eliminate the useless purchasing software, this process performs the best result comparing to other methods. On the other hand, software presentation produces 60-70% used in every user acceptance factors. This process has weak points and one of these weak points is that the organizations could not test the purchasing software before determining to purchase which increase opportunities of failures. The second rank of the risk management methods in every factor is the product prototype analysis which shows 80-90% of user satisfaction rates. This process examines the purchasing software through their current systems which confirms the compatibility and functions.

Based on Figure 9, the third performance risk management process is requirement; it produces 70-80% of the user satisfaction rate in every factor. This standard method that presents the moderate user acceptance rate due to there are various tools that might be used to extract users requirements. However, the defect of this method occurs when applying incorrect tools. So, the possibilities of failure in software purchasing will be increase. The risk management method in learning software behaviors and vendors' services from other organizations produce 70-85% user perception attitude. This process could guarantee the services after purchased and problems' occurrences. Software demonstration process shows 60-70% user satisfaction level. Since there is similarity between software demonstration and software presentation, the user acceptance rate of both processes are also similar.

From literature review [32][36][37], there are similarity of the internal processes of the POC, and requirement analysis in some factors which means there is not necessary to research on combinations of these two processes. From Figure 9 it can be concluded that POC, product analysis, requirement analysis, learning from other organizations, software demonstration and software presentation are results on user appreciate rate from high to low respectively.

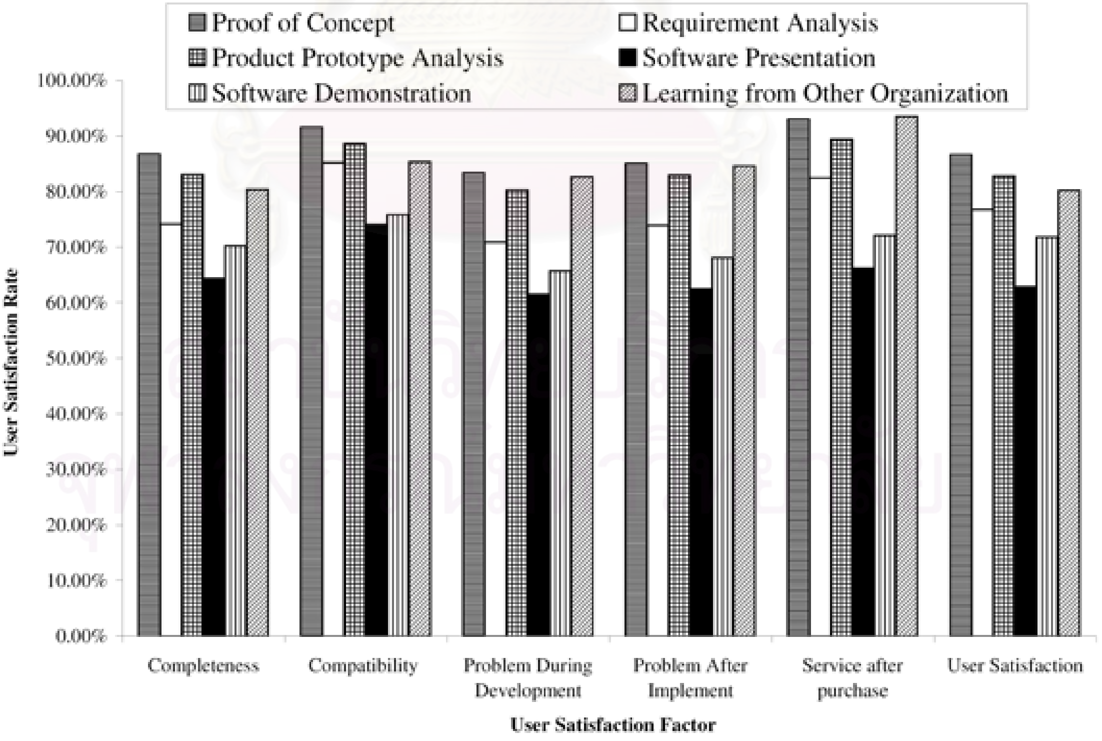


Figure 10: User Satisfaction Rate on Risk Management Process with Application-Based Organizations

Figure 10 presents the user acceptance rate of the application-based organizations comparing with user satisfaction factors of each selected risk management procedure. The results shown in Figure 10 is consistent with the results presented in Figure 9. This means that the consequences of user perceptions on risk management processes are going in the same direction as computer-based organizations. The POC still presents higher user satisfaction rate than the standard process.

Referring to Figure 10, except service after purchase factor, POC shows the highest rate in the user satisfaction. Moreover, the POC that applied in application-based organizations also presents 90% of users' satisfaction as same as computer-based organizations. The second rank in user perception attitude of risk management method is product prototype analysis which produces 80-90% of users' appreciation. However, the only difference between computer-based organization and application-based is in the third rank. Requirement analysis which shows 70-85% user acceptance level is lower performance than learning software behavior and vendor's services that shows 80-90% of users' perception. The lowest performance risk management method in application-based organizations is software presentation as same as computer-based organizations.

According to Figure 9 and Figure 10, it can conclude that POC provides the best user perception comparing to other standard processes.

Since the trend of the user satisfaction rate of the risk management process on the computer-based organization and application-based organization is going on the same way. Although, user satisfaction rate of application-based organizations is lower than the computer-based organizations. However, the user satisfaction rate on the risk management process that applied to purchase software between both organizations is still going on the same way. So, the rest will not be mentioned on the differences between computer-based and application-based organizations.

4.6 User Satisfaction Rate on the combination of two risk management processes and standard method

Some organizations use more than one risk management process to increase percentage of user satisfaction by combining between two the risk management processes which are proof of concept and product prototype analysis, and proof of concept and learning software behaviors and vendor’s services. The percentage of user perceptions on combining between POC and other 2 management methods which are learning software behaviors and vendors’ services from other organizations and product prototype analysis are displayed on Figure 11. According to the lower users’ expectation rate on software demonstration and software presentation, this thesis will not state on these procedures.

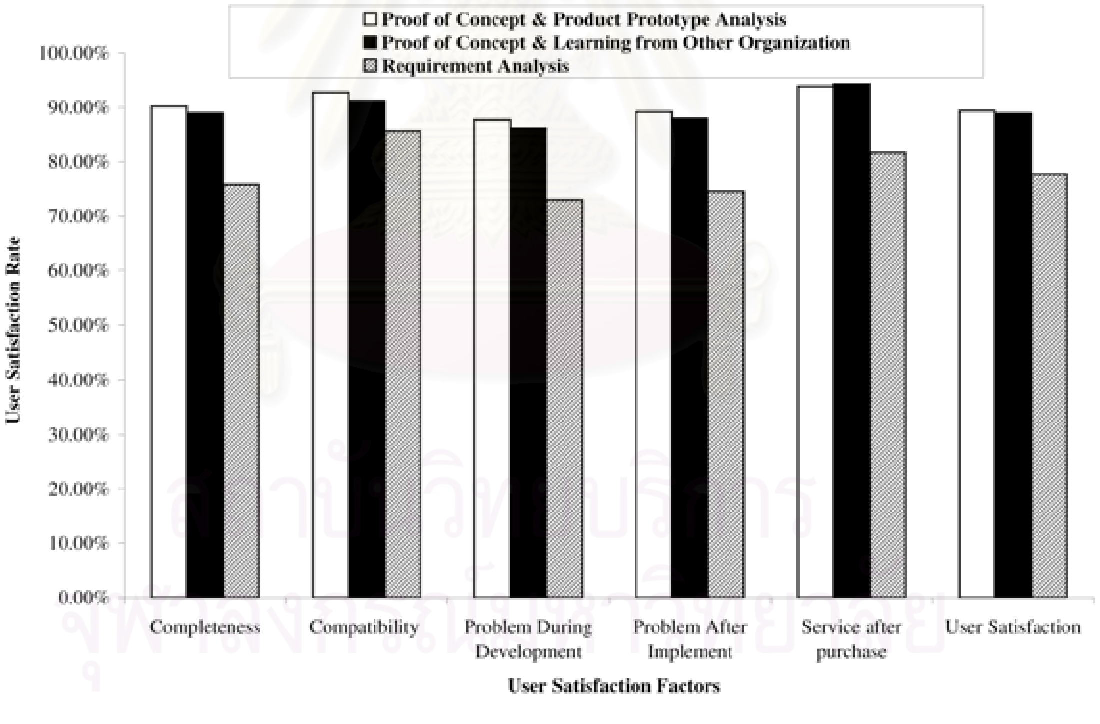


Figure 11: User Satisfaction Rate on the combinations of risk management processes and standard method

Figure 11 shows the user satisfaction rate on the combinations of risk management processes and standard methods. This figure shows the user satisfaction rates of combinations between two risk management processes: POC and product prototype analysis (90-95%), and POC and learning software behaviors and vendors' services from other organizations (88-92%). On the other hand, the users' acceptance rate of the standard method is 70-80%. Thus, it is clearly seen that the proposed method which obtained from the combination between POC and product prototype analysis significantly increase users' acceptance rate comparing with standard method.

According to Figure 11, the combination of product prototype analysis can produce highest user acceptance rate, except the case of the service after purchase. This is because all the weak points in the POC have been filled in by the product prototype analysis procedure. Additionally, since the POC is just the theoretical idea of the software, the procedure of the product prototype analysis helps confirming those concepts by analyzing the proposed software directly.

4.7 Proposed Method

From the previous section, the user satisfaction rates on combinations between two risk management methods may not be the final answer of this thesis. This research proposes the method by combining three risk management processes, including POC, product prototype analysis, learning software behaviors and vendors' services from other organizations. The combinations are based on the consideration of all weak points of every method. For evaluating the focusing methods, the questionnaire has distributed to 118 organizations as same organizations as previous collecting information. This questionnaire requires 6 months for collecting the user satisfaction rates in these methods. So, the organizations have chance to apply these methods and evaluate them. The results of the user acceptance level will shows in the next section.

4.8 Evaluation Results

The evaluation results are using the same formula as presented in previous. The result of the user satisfaction according to the focused methods comparing with the previous methods above is shown in Figure 12.

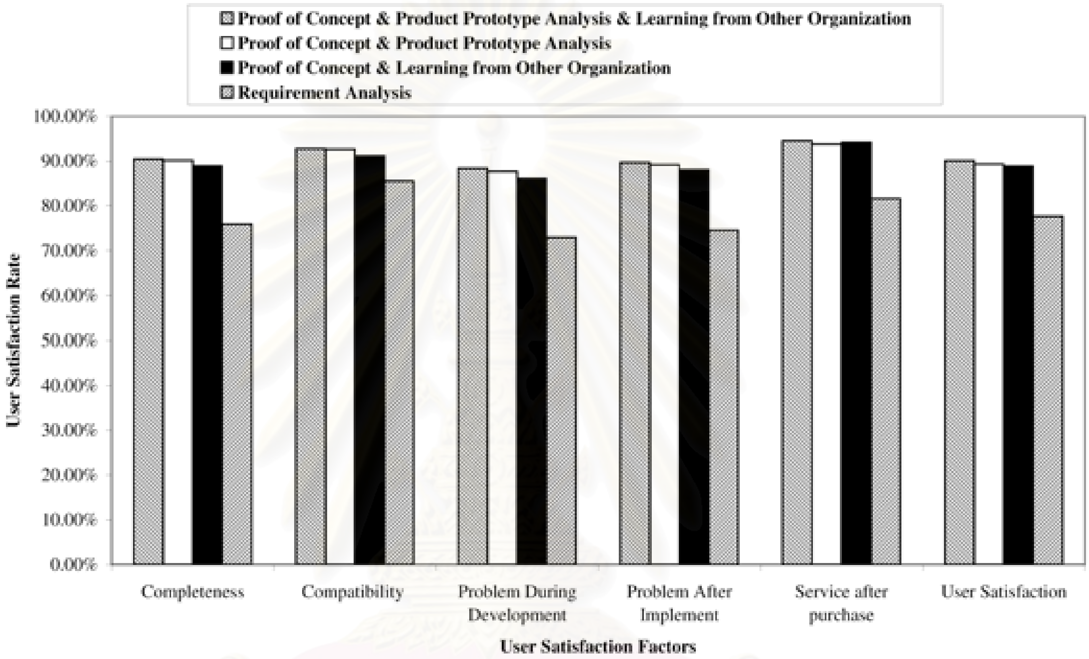


Figure 12: Comparing result between proposed methods against standard procedure

Figure 12 presents comparisons of user satisfaction rate among focusing method against standard procedures. As a result, combinations of various risk management procedures, including POC, product prototype analysis, and learning software behavior and vendors' services from other organizations produce 90-95% of the user acceptance rate which is the highest in every user satisfaction factor comparing with others. However, take more effort, budget and produce insignificant increasing number of user perception rate comparing with others focusing methods. The focusing methods include POC and product prototype analysis, which shows 90-95% of the user acceptance rate, and POC and learning software behaviors and vendors' services from other organizations, which shows 88-92% of the user perception rate. However, the user satisfaction rate of combinations of various risk management processes is significantly higher than the standard methods (70-80%). In order to reduce processes, work efforts, and total budgets in the software purchasing procedure, this thesis recommends that applying combinations between two risk managements, especially POC and product prototype analysis, obtains a better result than combinations of various risk management methods.

Next chapter describes the conclusions of this thesis discussion, and future works.

CHAPTER V

CONCLUSIONS AND FUTURE WORKS

This chapter discusses the outcome of this research comparing with all previous methods that have been implemented currently. In addition, Section 5.2, the conclusion of this thesis is described in details, and Section 5.3 is the further study of the remaining problems that should be solved for all kinds of organizations.

5.1 Discussion

Since the business world is highly competitions, every organization needs to maintain their abilities to be the leader of their business path. Nevertheless, the legacy software could not serve their needs in dynamically requirements. Moreover, the purchasing software process becomes a critical issue that organizations have to perform in vigilance. However, the current risk management methods that handle the software purchasing processes could not serve as required. So, the failure of implemented purchasing software could cause damage to the entire organization and may carry the profit of the company away.

In success criteria to improve risk managements in software purchasing, this thesis proposed the combinations between two risk management processes: the POC and product prototype analysis. The results show that the user satisfaction rate on this process significantly increases comparing with other risk management processes as shows in the previous Chapter. Each process of the combination can be used concurrently during the software selection process and supported each others to increase the user acceptance level. However, this process takes time to execute much higher than using only one risk management procedure while the outcome increases the accuracy of the purchasing software process.

Considering other factors, the proposed method turns out to have a higher user satisfaction rate while other risk management processes produce significantly less effective. Additionally, the proposed procedure uses a large amount of resources to perform various tasks in order to complete overall processes. Since the critical issue of the purchasing software is mostly concerning factor of the business organizational leadership.

As the result shows above, this thesis recommends organizations to apply two risk management processes: the POC and product prototype analysis. This combination method performs the best result comparing with the effective, work effort and spending costs rather than other risk management methods.

5.2 Conclusions

Currently, Thai's companies are relied on software to manage their businesses. Most software are obtained from the purchasing procedures. Unfortunately, the standard risk management method in the software purchasing process could not serve user requirements. Thus, the software purchasing process becomes critical issues because its results affect the entire operations of the organizations. Therefore, Thai's organizations need to find the better procedures to apply in the software purchasing process.

There are several risk management procedures that could be applied for handle software purchasing, including POC, product prototype analysis, software presentation, software demonstration, and learning from other organizations. Referring to the survey in this research, the standard risk management procedure shows the moderate user satisfaction and presents that the user perception is lower than the POC and product prototype analysis, as well as the combination of various methods. The previous results analyze that using only one risk management process could help organizations in selecting software with low capability. Although the combination of various risk management processes could solve the software purchasing process, the user satisfaction rate of this process does not significantly increase, comparing with the proposed method. In order to obtain accuracy on software purchasing process, organizations should use the proposed procedure which is the combination of two risk management processes, POC and product prototype analysis, to select the appropriate software.

5.3 Future Works

Although the proposed method has effectiveness in serving organizations in software purchasing process for right delivered software, there are some defects that should be eliminated for higher performance. According to the proposed processes, a large volume of assets or resources have to be applied in the purchasing procedures. Moreover, the processes are very rigid and time consuming. Therefore, these processes are not flexible and may not be suitable in some situations. Therefore, the future works that should be researched are as follows.

1. Apply concurrent methods to reduce the time. The concurrent method could reduce the project timeline by parallel the workload of the system; thus, some tasks can be completed earlier.
2. Consider the adaptation of the suggested method to apply the reusable concepts. Base on these concepts, the resources could be reused as needed and reduced timing that would be spent in reallocating the resources.

REFERENCES

[1] J.E. Bailey, and S.W. Pearson, Development of a Tool for Measuring and Analyzing Computer User Satisfaction. *Management Science*, V. 29, 5, (May 1983): 530-545.

[2] B. Ives, M.H. Olson and J.J. Baroudi, The Measurement of User Information Satisfaction. *Communication of the ACM*, V. 26, 10, (October 1983): 785-793.

[3] N.P. Melone, Theoretical Assessment of the User-Satisfaction Construct in Information Systems Research. *Management Science*, V. 36, 1, (Jan 1990): 76-91.

[4] A Risk Management Standard. *AIRMIC, ALARM, IRM*: (2002).

[5] A. Weatherall and F. Hailstones Risk Identification and Analysis using a Group Support System (GSS). *Proceedings of the 35th Hawaii International Conference on System Sciences* – (2002)

[6] Zhen-Yu Zhao, Lin-Ling Duan, An integrated risk management model for construction projects. *Management of Engineering & Technology, 2008. PICMET 2008. Portland International Conference on* (27-31 July 2008): 1389 – 1394.

[7] A.P. Sage, Risk management systems engineering. *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on V. 2,* (22-25 October 1995): 1033 – 1038.

[8] R. Buhr, A. Nel, M. dos Santos, Enterprise Risk Management: A New Philosophy. *Engineering Management Conference, 2006 IEEE International* (17-20 September 2006): 351 – 355.

[9] Syaripah Ruzaini, Syed Aris, Noor Habibah Arshad, Mohamed, Azlinah, Risk management practices in IT outsourcing projects. *Information Technology, 2008. ITSim 2008. International Symposium on V. 4,* (26-28 August 2008): 1 – 8.

[10] G.G. Roy, A risk management framework for software engineering practice. *Software Engineering Conference, 2004. Proceedings. 2004 Australian* (2004): 60 – 67.

[11] J. Nyfjord, M. Kajko-Mattsson, Software risk management: practice contra standard models. *Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on* (3-6 June 2008): 65 – 70.

[12] J. Ropponen, K. Lyytinen, Components of software development risk: how to address them? A project manager survey. *Software Engineering, IEEE Transactions on V. 26, 2,* (Feb. 2000): 98 – 112.

[13] M.F. Rabbi, K.O. Bin Mannan, A Review of Software Risk Management for Selection of Best Tools and Techniques. *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference on* (6-8 Aug. 2008):. 773 – 778.

[14] T.M. Skelton, H.J. Thamhain, A Stakeholder Approach to Minimizing Risks in Complex Projects. *Technology Management for the Global Future, 2006. PICMET 2006 V. 5,* (July 2006): 2203 – 2208.

[15] R.P. Higuera, Haimes, Y. Yacov, Software Risk Management. *(CMU/SEI-96-TR-012). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University,* 1996.

[16] F.J. Sisti, S. Joseph, Software Risk Evaluation Method. *(CMU/SEI-94-TR-19). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University*, (1994).

[17] B.W. Boehm, Software risk management: principles and practices. *IEE Software V. 8, 1,* (January 1991): 32 – 41.

[18] M.F. Rabbi, K.O. Bin Mannan, A Review of Software Risk Management for Selection of Best Tools and Techniques. *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference on* (6-8 Aug. 2008): 773 – 778

[19] R.P. Higuera, A.J. Dorofee, J.A. Walker, R.C. Williams, Team Risk Management: A New Model for Customer-Supplier Relationship. *(CMU/SEI-94-SR-005, ADA283987). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University,* (1994).

[20] W.W. Lowrance, Of Acceptable Risk. *Science and the Determination of Safety. Los Altos, Ca: William Kaufmann,* (1976).

[21] C. Chittister, H. Chittister, Y. Yacov, Risk Associated with Software Development: A Holistic Framework for Assessment and Management. *IEEE Transactions on Systems, Man, and Cybernetics V. 23, 3 (May-June1993)*: 710-723.

[22] T.C. Moore, A.B. Whinston, A Model of Decision Making with Sequential Information Acquisition. *Decision Support Systems, Vol. 2, 4,* (1986): 289-308.

[23] W. Scacchi, Process Models in Software Engineering. *Institute for Software Research, University of California, Irvine.* (February 2001).

[24] Kirkpatrick, R.J. Walker, J. Robert, F. Robert, Software Development Risk Management: An SEI Appraisal. *Software Engineering Institute Technical Review '92 (CMU/SEI-92-REV). Pittsburgh, Pa.: Software Engineering Institute, Carnegie Mellon University,* (1992).

[25] T.H. Devenport, J.E. Short, The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review,* (1990): 11-27.

[26] Zao-Bin Gan, Deng-Wen Wei, Jin-Long Zhang, V. Varadharajan, Business-process-oriented software requirements automatic generator. *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on V. 1*, (4-7 July 2005): 95 – 98.

[27] M. Hammer, J. Champy, Reengineering the Corporation: A Manifesto for Business Revolution. *N. Brealey, London,* (1993).

[28] M.J. Earl, The New and the Old of Business Process Redesign. *Journal of Strategic Information Systems, V. 3, 1,* (1994): 5-22

[29] A.O. Adebowale, Systems Development Requirement analysis Process. *Computer Resource Center Federal University of Technology*, *Akure* (July 2008).

[30] I. Miyamoto, R.T. Yeh, A software requirement analysis and definition methodology for business data processing. *AFIPS Joint Computer Conferences Proceedings of the* (4-7 May 1981): 571-580

[31] M. Jackson, Software Requirements & Specifications: A Lexicon of Practice, Principles and Prejudices. *Addison-Wesley*, (1995).

[32] H. Sharp, A. Finkelstein, G. Galal, Stakeholder Identification in the Requirements Engineering Process. *Workshop on Requirements Engineering Processes (REP'99) - DEXA'99, Florence, Italy,* (1-3 September 1999): 387-391

[33] J. Vanwelkenhuysen, Quality requirements analysis in customer-centered software development. *Requirements Engineering, 1996., Proceedings of the Second International Conference on* (15-18 April 1996) 117 – 124

[34] P.G. Ranky, R.J. Caudill, K. Limaye, N. Alli, S. ChamyVelumani, A. Bhatia, M. Lonkar, A disassembly user requirements analysis method, tools and validated examples. *Electronics and the Environment, 2002 IEEE International Symposium on* (6-9 May 2002): 128 – 132

[35] J. Mylopoulos, L. Chung, S. Liao, H. Wang, E. Yu, Exploring alternatives during requirements analysis. *IEEE Software V. 18, 1,* (January-February 2001): 92 – 96

[36] R. Rudd, Should Your Software Selection Process Have a Proof of Concept? Part One : Structures and the Selection Process. (July 2004).

[37] R. Rudd, Should Your Software Selection Process Have a Proof of Concept? Part Two : Advantages, Disadvantages and Conclusion (July 2004).

[38] G.T. Clay, and P.G. Smith, Rapid Prototyping Accelerates the Design Process. *Machine Design.* (9 March 2000).

[39] T. Edmark, Model Approach: Creating a Prototype that will Wow Investors. *Entrepreneur.* (October 1997).

[40] M. Schrage, How Prototypes Can Change Your Business. *Across the Board.* (January 2000).

[41] C. A. Gulbranson Proof of Concept Centers: Accelerating the Commercialization of University Innovation. *Ewing Marion Kauffman Foundation,* (January 2008).

[42] B. A. Nuseibeh and S. M. Easterbrook**,** Requirements Engineering: A Roadmap. *In A. C. W. Finkelstein The Future of Software Engineering IEEE Computer Society Press*, (2000).

[43] A. Davis, Operational Prototyping: A New Development Approach. *Software*, (9 May 1992): 70-78.

[44] R. Yamamoto, K. Yamamoto, K. Ohashi, J. Inomata, Development of a business process modeling methodology and a tool for sharing business processes. *Software Engineering Conference, 2005. APSEC '05. 12th Asia-Pacific* (15-17 Dec. 2005): 8.

[45] R. Hosseini, A practical approach for measuring IT-support of business processes. *Computer and Information Technology, 2005. CIT 2005. The Fifth International Conference on* (21-23 September 2005): 1099 – 1103.

[46] L.S. Globa, T.N. Kot, V.N Oleniuk, Environment for system prototype analysis and testing. *Microwave & Telecommunication Technology, 2008. CriMiCo 2008. 2008 18th International Crimean Conference* (8-12 September 2008): 382 – 383.

[47] T. Makinen, T. Varkoi, H. Jaakkola, Assessment of a software process assessment process. *Management of Engineering and Technology, 2001. PICMET '01. Portland International Conference on V. 1*, (29 July-2 Aug. 2001): 437

[48] M. Verlage, Towards software process modules. *Software Process Workshop, 1996. 'Process Support of Software Product Lines'., Proceedings of the 10th International* (17-19 June 1996): 112 – 114

[49] M. Ahmed-Nacer, Towards a new approach on software process evolution. *Computer Systems and Applications, ACS/IEEE International Conference on. 2001* (25-29 June 2001): 345 – 351ม

APPENDIX

Questionnaire for Elicitation Process

แบบสอบถามนี้เป็นส่วนหนึ่งของงานวิทยานิพนธ์ ในหลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ (ภาคภาษาอังกฤษ) จุฬาลงกรณ์มหาวิทยาลัย โดยผลการศึกษาวิจัยที่ได้ จะนำไปใช้เพื่อบริหารจัดการความเสี่ยงในการจัดซื้อซอฟต์แวร์ ขององค์กร เพื่อเป็นแนวทางในการพัฒนาและการวางแผนการตลาด เพื่อตอบสนองความต้องการขององค์กรได้อย่างมีประสิทธิภาพ โดยข้อมูลที่ท่านได้ตอบแบบสอบถามนี้ ผู้ศึกษาจะถือเป็นความลับ และไม่นำไปเปิดเผย

ตอนที่ 1 ข้อมูลเกี่ยวกับปัจจัยส่วนบุคคล

คำชี้แจง โปรดทำเครื่องหมาย / ลงในช่อง ( ) หน้าข้อความที่ตรงกับความคิดเห็นของท่านมากที่สุด

1. เพศ

1. ( ) ชาย 2. ( ) หญิง

2. ระดับการศึกษา

1. ( ) มัธยมศึกษา 2. ( ) อนุปริญญา

3. ( ) ปริญญาตรี 4. ( ) ปริญญาโท

5. ( ) สูงกว่าปริญญาโท 6. ( ) อื่นๆ.................................

3. อายุ

1. ( ) 21-25 ปี 2. ( ) 26-30 ปี

3. ( ) 31-35 ปี 4. ( ) 36-40 ปี

5. ( ) 41-45 ปี 6. ( ) 46-50 ปี

7. ( ) 51 ปีขึ้นไป 8. ( ) อื่นๆ.......................................

4. ตำแหน่ง

1. ( ) พนักงานทั่วไป 2. ( ) System Analyst

3. ( ) Project Manager 4. ( ) Assisted Vice President

4. ( ) Vice President 5. ( ) CEO

6. ( ) อื่นๆ...

5. ประเภทของธุรกิจ

1. ( ) การเงินและธนาคาร 2. ( ) Information Technology

3. ( ) หน่วยงานราชการ 4. ( ) อุตสาหกรรม

5. ( ) ธุรกิจนำเข้า-ส่งออก 6. ( ) คมนาคมขนส่ง

7. ( ) สื่อสาร 8. ( ) บันเทิง

9. ( ) อสังหาริมทรัพย์ 9. ( ) อื่นๆ ....

6. ขนาดของธุรกิจ

1. ( ) เล็ก 2. ( ) กลาง 3. ( ) ใหญ่

ตอนที่ 2.ข้อมูลพฤติกรรมการจัดซื้อซอฟต์แวร์

คำชี้แจง โปรดทำเครื่องหมาย / ลงในช่อง ( ) หน้าข้อความที่ตรงกับความคิดเห็นของท่านมากที่สุด

1. องค์กรของท่านเคยซื้อซอฟต์แวร์จากผู้ขาย เพื่อนำมาใช้กับองค์กรหรือไม่

1. ( ) เคย 2. ( ) ไม่เคย

2. เหตุผลหลักในการตัดสินใจเลือกซื้อซอฟต์แวร์จากผู้ขายเพื่อนำมาใช้กับองค์กร (ตอบได้มากกว่า 1 ข้อ)

1. ( ) เพิ่มมูลค่าการตลาด
2. ( ) ขยายการเจริญเติบโตทางธุรกิจ
3. ( ) ให้เป็นมาตรฐานเดียวกัน
4. ( ) เพิ่มความน่าเชื่อถือ
5. ( ) เพื่อการแข่งขันกันทางธุรกิจ
6. ( ) เพิ่มประโยชน์ใช้สอย
7. ( ) ความเหมาะสมกับระบบเดิม
8. ( ) ราคาและบริการหลังการขาย
9. ( ) ความทันสมัย
10. ( ) อื่นๆ.................................

3. วิธีการหาแหล่งที่มาของผู้ขาย (ตอบได้มากกว่า 1 ข้อ)

1. ( ) ป้ายโฆษณา
2. ( ) Internet
3. ( ) นิตยสาร
4. ( ) มีความสัมพันธ์กับองค์กร
5. ( ) ยื่นซองประมูล
6. ( ) อื่นๆ......................................

4. ปัจจัยสำคัญที่มีอิทธิพลต่อการเลือกซื้อซอฟต์แวร์ (ตอบได้มากกว่า 1 ข้อ)

1. ( ) คุณสมบัติประโยชน์ใช้สอย
2. ( ) ราคา
3. ( ) ค่าบำรุงรักษา
4. ( ) บริการหลังการขาย
5. ( ) ชื่อเสียงของผู้ขาย

5. ( ) อื่นๆ ....

5. นโยบายการในการเลือกซื้อซอฟต์แวร์ (ตอบได้มากกว่า 1 ข้อ)

1. ( ) ความเหมาะสมกับองค์กร
2. ( ) สนับสนุนการทำงานในปัจจุบัน
3. ( ) ทันสมัย
4. ( ) สามารถแข่งขันกับคู่แข่ง
5. ( ) ราคา
6.( ) ตรงความต้องการมากที่สุด
7. ( ) ของแถม
8. ( ) บริการหลังการขาย
9. ( ) ของฟรีนำมาพัฒนาใช้เอง
10. ( ) อื่นๆ ....

6. กระบวนการตัดสินใจเลือกซื้อซอฟต์แวร์ จากผู้ขาย เพื่อให้ตรงกับความต้องการขององค์กร (ตอบได้มากกว่า 1 ข้อ)

1. ( ) Proof of Concept 2. ( ) Requirement Analysis

3. ( ) Product Prototype Analysis 4. ( ) Software Presentation

5. ( ) Software Demonstration 6. ( ) ศึกษาจากองค์กรอื่นที่เคยใช้งาน

ผลิตภัณฑ์ 7. ( ) อื่นๆ......................................

7. จากกระบวนการคัดสรรในการเลือกซื้อซอฟต์แวร์ดังกล่าวข้างต้น กรุณาทำเครื่องหมาย (/) ในช่องว่างตามความพึงพอใจที่ได้รับหลังจากกาซื้อซอฟต์แวร์

(1=**ไม่พอใจมาก**, 2=**ไม่พอใจ**, 3=**เฉยๆ**, 4=**พึงพอใจมาก**,5=**พึงพอใจมากที่สุด**)

	1	2	3	4	5
1) ซอฟต์แวร์ที่ซื้อทำงานได้ตรงกับความต้องการ					
2) ความเหมาะสมกับองค์กร					
3) คุณภาพเหมาะสมกับราคา					
4) คุณสมบัติครบถ้วนตามที่ผู้ผลิตนำเสนอ					
5) ปัญหาระหว่างการพัฒนา					
6) ปัญหาระหว่างการใช้งาน					
7) การบริการหลังการขาย					
8) ภาพลักษณ์ขององค์กร					
9) ความพึงพอใจขององค์กร					
10) ความพึงพอใจของผู้ใช้งาน					
11) ความน่าเชื่อถือขององค์กร					

8. นโยบายรองรับเมื่อซอฟต์แวร์ ที่ซื้อไม่ตรงความต้องการ (ตอบได้มากกว่า 1 ข้อ)

1. ( ) เปรียบเทียบปรับตามที่ระบุในสัญญา

2. ( ) นำไปใช้กับหน่วยงานอื่นที่มีความต้องการคล้ายคลึงกัน

3. ( ) เก็บไว้ใช้ศึกษาหรือรอเวอร์ชันปรับปรุง

4. ( ) คืนผลิตภัณฑ์ที่ไม่ตรงความต้องการ

5. ( ) พัฒนาซอฟต์แวร์ ขึ้นเพื่อใช้เอง

6. ( ) อื่นๆ...

9. ข้อเสนอแนะอื่นๆ

....

....

....

Questionnaire for Evaluation Process

แบบสอบถามนี้เป็นส่วนหนึ่งของงานวิทยานิพนธ์ ในหลักสูตรวิทยาศาสตร์มหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ (ภาคภาษาอังกฤษ) จุฬาลงกรณ์มหาวิทยาลัย โดยผลการศึกษาวิจัยที่ได้ จะนำไปใช้เพื่อบริหารจัดการความเสี่ยงในการจัดซื้อซอฟต์แวร์ ขององค์กร เพื่อเป็นแนวทางในการพัฒนาและการวางแผนการตลาด เพื่อตอบสนองความต้องการขององค์กรได้อย่างมีประสิทธิภาพ โดยข้อมูลที่ท่านได้ตอบแบบสอบถามนี้ ผู้ศึกษาจะถือเป็นความลับ และไม่นำไปเปิดเผย

ทั้งนี้ผู้ศึกษาได้ผลลัพธ์จากการทำการวิจัยในเรื่องกระบวนการบริหารจัดการความเสี่ยงในการจัดซื้อซอฟต์แวร์ขององค์กร โดยมีกระบวนการบริหารจัดการความเสี่ยงดังนี้

1. Proof of Concept
2. Product Prototype Analysis
3. Learning Software Behaviors and Vendors' Services from Other Organizations

เพื่อให้ได้ผลลัพธ์ที่แม่นยำถูกต้องและแน่นอนจึงใคร่ขอสอบถามความคิดเห็นขององค์กรในเรื่องกระบวนการบริหารจัดการความเสี่ยงในการจัดซื้อซอฟต์แวร์ โดยใช้ระยะเวลาในการตัดสินใจในช่วง 3-6 เดือนหลังจากทดลองใช้หรือมีข้อแนะนำใดๆ เรียนแจ้งให้ทราบจักขอบพระคุณอย่างสูง

จากกระบวนการบริหารจัดการความเสี่ยงในการเลือกซื้อซอฟต์แวร์ดังกล่าวข้างต้น กรุณาทำเครื่องหมาย(⁄) ในช่องว่างตามความพึงพอใจที่ได้รับหลังจากกาซื้อซอฟต์แวร์

(1=**ไม่พอใจมาก**, 2=**ไม่พอใจ**, 3=**เฉย ๆ**, 4=**พึงพอใจมาก**, 5=**พึงพอใจมากที่สุด**)

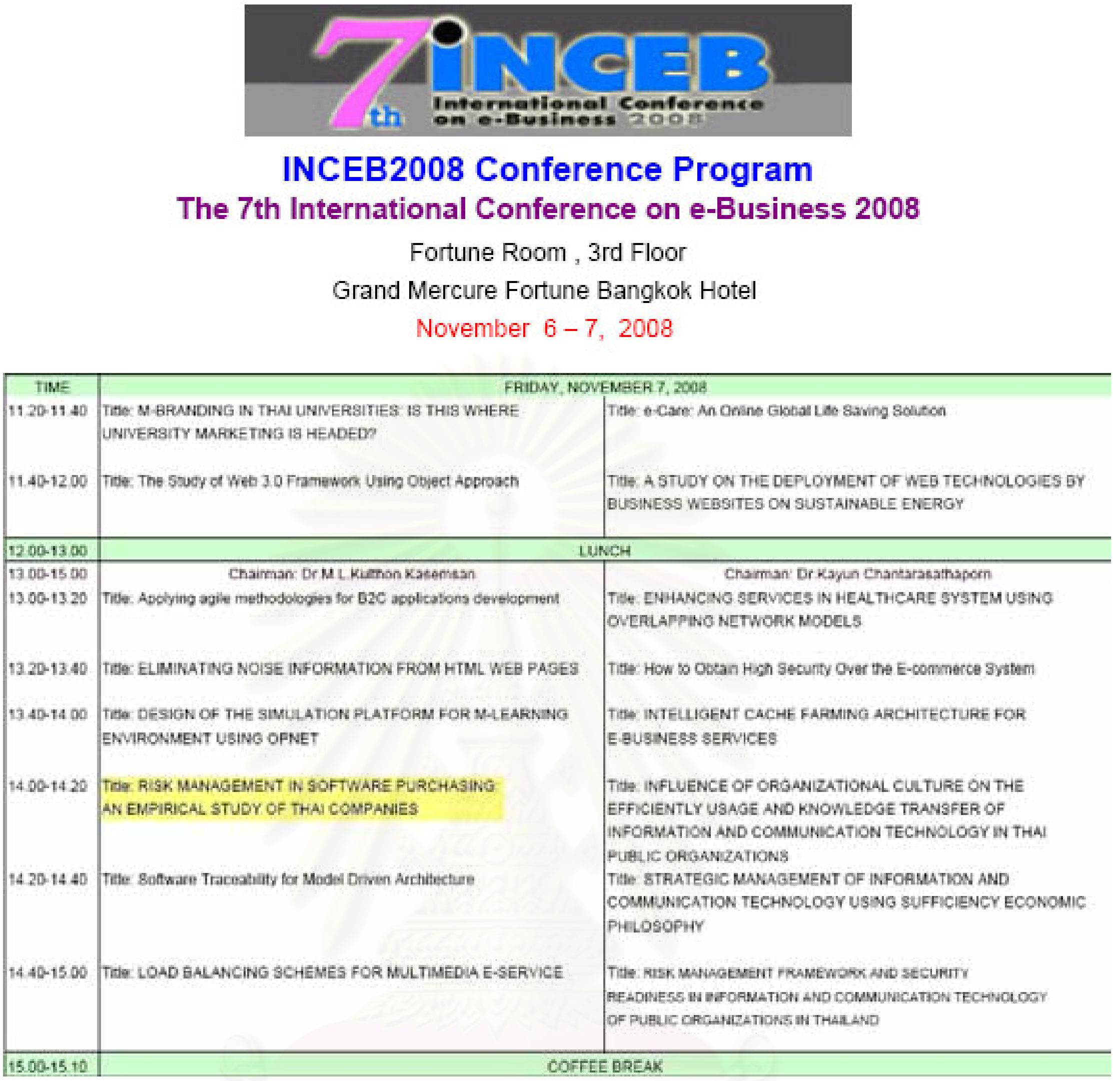
	1	2	3	4	5
1) ซอฟต์แวร์ที่ซื้อทำงานได้ตรงกับความต้องการ					
2) ความเหมาะสมกับองค์กร					
3) คุณภาพเหมาะสมกับราคา					
4) คุณสมบัติครบถ้วนตามที่ผู้ผลิตนำเสนอ					
5) ปัญหาระหว่างการพัฒนา					
6) ปัญหาระหว่างการใช้งาน					
7) การบริการหลังการขาย					
8) ภาพลักษณ์ขององค์กร					
9) ความพึงพอใจขององค์กร					
10) ความพึงพอใจของผู้ใช้งาน					
11) ความน่าเชื่อถือขององค์กร					

ข้อเสนอแนะอื่นๆ

....

....

....



7th INCEB International Conference on e-Business 2008

INCEB2008 Conference Program

The 7th International Conference on e-Business 2008

Fortune Room , 3rd Floor

Grand Mercure Fortune Bangkok Hotel

November 6 – 7, 2008

TIME	FRIDAY, NOVEMBER 7, 2008	
11.20-11.40	Title: M-BRANDING IN THAI UNIVERSITIES: IS THIS WHERE UNIVERSITY MARKETING IS HEADED?	Title: e-Care: An Online Global Life Saving Solution
11.40-12.00	Title: The Study of Web 3.0 Framework Using Object Approach	Title: A STUDY ON THE DEPLOYMENT OF WEB TECHNOLOGIES BY BUSINESS WEBSITES ON SUSTAINABLE ENERGY
12.00-13.00	LUNCH	
13.00-15.00	Chairman: Dr M.L.Kulthon Kasemsan	Chairman: Dr Kayun Chantarasathaporn
13.00-13.20	Title: Applying agile methodologies for B2C applications development	Title: ENHANCING SERVICES IN HEALTHCARE SYSTEM USING OVERLAPPING NETWORK MODELS
13.20-13.40	Title: ELIMINATING NOISE INFORMATION FROM HTML WEB PAGES	Title: How to Obtain High Security Over the E-commerce System
13.40-14.00	Title: DESIGN OF THE SIMULATION PLATFORM FOR M-LEARNING ENVIRONMENT USING OPNET	Title: INTELLIGENT CACHE FARMING ARCHITECTURE FOR E-BUSINESS SERVICES
14.00-14.20	Title: RISK MANAGEMENT IN SOFTWARE PURCHASING: AN EMPIRICAL STUDY OF THAI COMPANIES	Title: INFLUENCE OF ORGANIZATIONAL CULTURE ON THE EFFICIENTLY USAGE AND KNOWLEDGE TRANSFER OF INFORMATION AND COMMUNICATION TECHNOLOGY IN THAI PUBLIC ORGANIZATIONS
14.20-14.40	Title: Software Traceability for Model Driven Architecture	Title: STRATEGIC MANAGEMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY USING SUFFICIENCY ECONOMIC PHILOSOPHY
14.40-15.00	Title: LOAD BALANCING SCHEMES FOR MULTIMEDIA E-SERVICE	Title: RISK MANAGEMENT FRAMEWORK AND SECURITY READINESS IN INFORMATION AND COMMUNICATION TECHNOLOGY OF PUBLIC ORGANIZATIONS IN THAILAND
15.00-15.10	COFFEE BREAK	

RISK MANAGEMENT IN SOFTWARE PURCHASING: AN EMPIRICAL STUDY OF THAI COMPANIES

S. Yenyuak[1] *and P. Bhattarakosol*[1]

setaphat.y@student.chula.ac.th and pattarasinee.b@chula.ac.th

[1]Department of Mathematics, Faculty of Science,

Chulalongkorn University, Thailand

Abstract

When business competition is increased, it is possible that legacy software could not serve the response on business requirements. Software purchasing is the solution that could help organizations to take action on dynamically requirements. Some organizations have been successfully implemented newly software while some fail according to the differences between risk management processes that selected from each organization. Therefore, a survey of risk management processes in various business organizations in Thailand had been performed and proposed the most accurately risk management factors that could reduce percentage of failure comparing with organizations' satisfaction rate. Additionally, the experimental results have shown that applying standard risk management process is not meet user satisfaction while some risk management processes of Thai's organizations performed better results rather than standard.

Index Terms— user satisfaction, risk management process, satisfaction rate, standard software purchasing

1. INTRODUCTION

There is no standard for purchasing software process. Many organizations define their own processes to filter software that meet their requirements. According to this, a number of organizations fail to achieve their new software. Sometimes the purchased software could not proper meet their requirements, some features cannot be used, incompatible with their system or culture, and not meet user satisfaction [1][2][3]. Thus, various risk management process have been proposed to protect and prevent the failure in software purchasing.

There are several risk management processes to apply for software selection, such as proof of concept [4][5], requirement analysis [6], software presentation, product prototype analysis [7][8][9], software demonstration, and learning from other organizations. However, the combination of these methods could reduce risk. Numerous studies have shown that over half of software development projects fail; the significant reason that causes the failure is breakdown in the "requirements elicitation" process. Additionally, problems occurred during software implementation are such as vendor services, maintenance costs, features and functions do not meet the demand, incompatible with system, fail user satisfaction and software liabilities, etc.

Referring to the requirement elicitation process, this process produces requirements of users that lead to the software purchasing. Therefore, the details obtained from this process are very significantly important. Since there are various tools to be applied in this elicitation process, such as use cases, and data flow diagrams, etc., these tools must be implemented properly by the software developer team, especially software analysts. Unfortunately, the final results that are the delivered software or purchased software, mostly, do not completely meet the needs of the entire organization.

Although there is failure in software purchasing in some organizations, applying risk management helps reducing this failure in other organizations as well. Therefore, this paper focuses in user satisfaction result of Thai companies on each risk management. The expected outcome from this paper is to present the currently procedures in software purchasing that every organization used.

This paper is organized as follows. Section 2 introduces the standard risk management process. Section 3 represents organization surveys method; data analysis and the results in Section 4. Discussion about standard risk management against Thai's risk management procedure has been described in Section 5. Finally, conclusions are drawn in Section 6.

2. STANDARD RISK MANAGEMENT PROCESS ON SOFTWARE PURCHASING

Standard risk management process [10] on software purchasing which widely use in every IT-based is based on the requirement analysis process. Due to the requirement analysis process is the process for understanding users' needs and expectations from a proposed system or application. Additionally, this process is the most important stage that must be well-defined in the Software Development Life Cycle model otherwise its failures will affect to all other following processes that cause the software failure to the entire organization thus overall process was adopted to fit in standard risk management process.

Requirements [6] are a description of how a system should behave or a description of system properties or attributes. It can alternatively be a statement of 'what' an application is expected to do.

Software Requirements Analysis Process covers the complex task of eliciting and documenting the requirements of all users, modeling and analyzing requirements and documenting as a basis for system design.

Proof of concept (POC) [4][5] requires an agreement set of proofs or tests that define its success criteria. Proof requirements are generally agreed and documented after the vision scope document is completed and before defined solution. This document defines the proof context and identifies individual components of proof requirements. Each proof includes a requirement, proof method, acceptance criteria, and success measures.

Proof requirements relate to the objectives and goals established in the vision scope document. In very short POC's, this document may be combined with the vision scope document.

Proof Requirements: The proof requirements are used to drive the POC solution design and to manage project scope during subsequent phases of the POC. It is used as both a reference and a driver for the proof presentation and documentation and can also be used to record results against each required proof.

Objectives and Scope: The POC objectives, scope, and overall solution vision are defined in the vision scope document. The proof requirements document defines how the solution vision will be confirmed by the POC. The proof requirements define at a specific level how this will be achieved.

Previous studied conclude that, the POC process normally occurs after software is selected. It is a major undertaking and should not be used as a tool to compare one solution over another. Furthermore, as the POC is orientated to the chosen solution and implementation methodology, the content or process undertaken cannot be readily transferred to another software solution. Using only POC could be used to confirm the preferred vendor's status rather than establishing it.

There are a number of advantages to the POC for the organization including:

Organization could meet expected synchronization with the vendor. In addition, organization could extract and identify software functional gaps or overselling. In each evaluation phase, evaluated members have to test and track software based on organization's requirement.

POC improved accuracy scoping which provided a better understanding of the organization requirement to complete the implementation. POC process allows the organization to evaluate the software and implementation vender.

However there are limitations of the POC which should be noted as follows.

Depending on the commercial agreements, the POC can be placed within the sales cycle thus software's vendor is restricted for fully disclosure. Further, the documentation produced within the POC may have marketing content which does not add value to the project.

A POC should be completed as part of the selection process when the risk of project failure is comparatively high. Risk can be measured by two key variables: complexity of requirements, and level of expertise within the selection team members.

Product prototype analysis [7][8][9] involves the production of functionally useful and trustworthy systems through experimentation with evolving systems. Generally, the experimentation is conducted with many users' involvement in the evaluation of the prototype.

Software demonstration displays software behavior, features and functions. This process can identify software weak points. This process could help organization in decision which software should proper for organization. It shows software feasibility and user friendly.

Software presentation shows software advantages strong points, feasibility, and functionality. It also presents software interface, software language support and minimum requirement of the system.

Risk management process on learning software behavior from other organizations researched for other organizations that used the purchasing software. In term of service, functionality, compatibility, usability and flexibility, that how many trouble during implementation and production.

Although, the standard risk management procedures have ability to eliminate risk on purchasing the worthless software, unfortunately each procedure also has disadvantage features. Therefore, applying the standard procedure does not guarantee that the delivered software will be completed as needed. Thus, in order to avoid the software failure, all business organizations have to determine their own standard risk management process that ensures their needs to be served. Therefore, this research will focus on the risk management process that the business organizations have really applied to support their satisfactions in software purchasing.

3. ORGANIZATIONAL SURVEY METHOD

In the business section, software failures can cause dramatically damage to the entire organization. Therefore, selecting software to be implemented is an important issue that every manager cannot ignore. Additionally, most business organizations have different business profiles and objectives. Thus, the software purchasing process for each organization is usually different from each other. So, this research, the study of software purchasing has been performed using depth interviewing, including distributing the questionnaires to various organizations with different sizes and different business objectives.

The interviewing surveys method is performed by walk in to organization and send questionnaire directly to the organizations. Sampling data was collected form IT-related-based business on both computer-based and application-based organizations, including all business classifications, such as hospitals, airlines, communications, information technologies, financial and banking, and etc.

The interviewing process has performed in 8 companies. These companies can be classified as financial, software consult and support, software house, and car rental companies. Moreover, the sizes of these companies can be categorized in 3 large companies, 2 medium companies and 3 small companies. Focusing on the quality of result, we have to interview management level person who have to make a decision on purchasing and have best knowledge on the whole project. Thus, every interviewee is in the rank of manager or president of the company who authorizes in software purchasing process of the organization; the interviewing time of each place is about 1 hour.

The business size can be classified into 3 groups includes large, medium and small defined by Thai’s ministry of industry. Organizations which have investment budget less than 5 million baht are defined to small. Organizations which have investment budget during 5-200 million baht can be defined to medium. Organizations which have investment budget greater than 200 million baht represent to large.

In order to obtain the real risk management criteria in the business area for software purchasing, another elicitation method to capture decision factors of business organization is the use of questionnaire. Questionnaires are distributed to 110 companies; these can be classified as 22 large, 71 medium, and 15 small sizes as same as the companies in the interviewing process.

4. SURVEY RESULTS

Thai's organization software purchasing process was collected from interviewed and questionnaire normally includes below software selection process which are:

Read advertisement on computer magazines, open vendor auction, or searched the internet on software ranking website.

Announce organization requirement and let software's vendor to give and auction.

Provided organization system overviewed to selected vendors. Organizations can analyze possibility to implement their software.

Set up software purchasing project team members. Team members include system analysts, developers, business analysts, and users.

Invited selected vendors to present their software features, implemented processes and plan.

Applied organization's risk management process for decision making on software purchasing.

Thai's organization typically used at least two of software selection processes above to obtain the purchased software.

The results from the elicitation process show that there are different risk management processes among different sizes of organizations, with irrelevant to the business profiles.

There are mainly two influence types that major concerned: business influencers, and technical influencers. It is obvious that the business size effects on the software purchasing process. By the business size, small business organizations usually avoid using complicated risk management process in case of reducing time and tasks, while the larger business size concentrates in complicated risk management processes because it could produces better result than ease tasks.

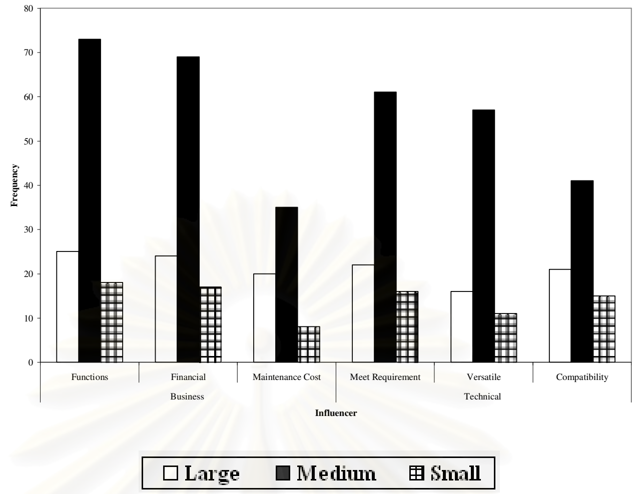


Figure 1: Consideration Factors for Software Purchasing of Each Company's Size

According to Figure 1, it is obvious that every organization concerns in financial, functions, the maintenance cost, requirement satisfactions, versatilities, and compatibilities. However, the large organizations, mostly, concern in financial, functions, meet requirements, and compatibilities more than any other influences. Medium organizations concern only in financial, functions, and requirements accomplishment. Small organizations are different from other sizes because it most concerns factor are financial, functions, and compatibility.

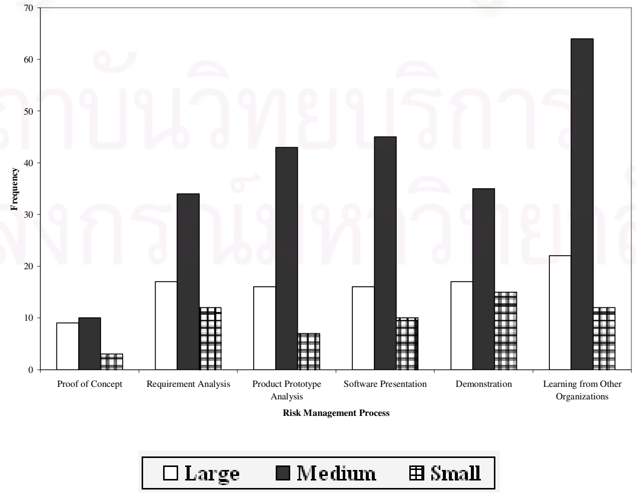


Figure 2: Risk Management Process for Software Purchasing

As the consequence of the result mentioned in the previous section, the most important factor that every companies must determine before the risk management starts are financial, functions, maintenance, compatibilities, and requirements fulfillment.

Figure 2 shows that the risk management process, which is widely used from Thai's companies, is learning from other organizations because it is the easiest way to obtain software behavior and vendor services without taking much effort. While POC represents the lowest used from the surveys. The standard risk management procedure which is requirement analysis is medium used on every business size.

The User Satisfaction Percentage = $\frac{\sum_{i=1}^{5}(N_i * i)}{M * 5} * 100$

N = Total Number of Organization that presents in each satisfaction level.

i = Satisfaction Level value = 1,2,3,4 and 5 greater means much satisfy.

M = Total Number of Organization uses each risk management process.

Figure 3 shows the user appreciates rate of computer-based organization using every focusing risk management method. POC demonstrates the highest user acceptance even though it is not widely used in many organizations because it has to perform more complicated tasks and take a lot of resources to be done. Software presentation shows the lowest user acceptance. Requirement analysis shows moderate user satisfaction rate. From Figure 3, there are similarity between POC, requirement analysis and product prototype analysis in some factors which means there is no difference on user expectations between selected procedures on these factors. It can be concluded that POC, product analysis, requirement analysis, learning from other organization, software demonstration and software presentation are results on user appreciate rate from high to low respectively.

Figure 4 presents user acceptance rate of application-based organization using selected risk management procedures. The results shown in Figure 4 is consistent with the results presented in Figure 3. This means that the consequences of user perceptions on risk management processes are going in the same way like computer-based organizations. POC still presents higher user satisfaction rate than standard process.

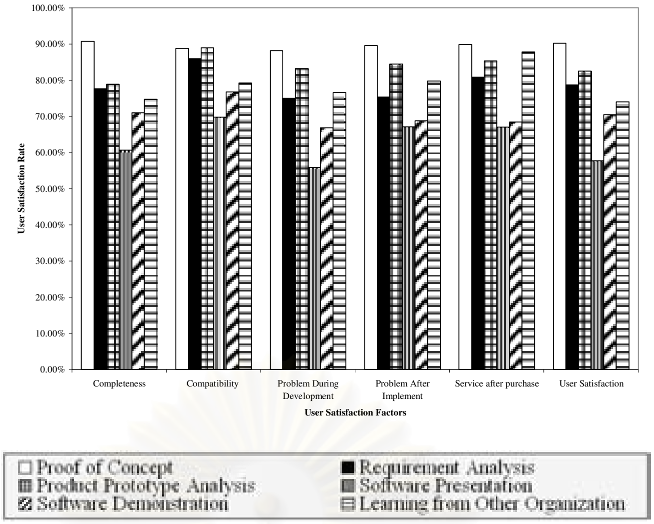


Figure 3: User Satisfaction Rate on Risk Management Process with Computer-Based Organization

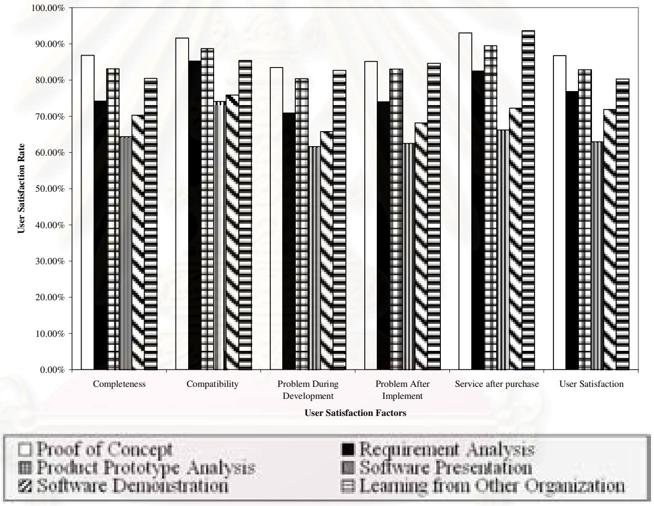


Figure 4: User Satisfaction Rate on Risk Management Process with Application-Based Organization.

According to Figure 3 and Figure 4, it can conclude that POC provides the better user perception than the standard process.

Since the trend of the user satisfaction rate of the risk management process on the computer-based organization and application-based organization is going on the same way. Although, application-based organization presents user satisfied rate lower than computer-based organization. However, the user satisfaction rate on the risk management process that applied to purchase software between both organizations is still going on the same way. So, the rest will not be mentioned on the differences between computer-based and application-based organizations.

Some organizations are using more than one risk management process to increase percentage of user satisfaction by combining between the risk management processes. The percentage of user perceptions on combining between POC and other 2 management methods which are learning form other organization and product prototype analysis will display on Figure 5.

Figure 5 shows that combinations between risk management processes significantly increase user acceptance rate. Additionally, it shows that the combinations present better user satisfactions than the standard software purchasing procedure. The user perception rates between these two methods are significantly different.

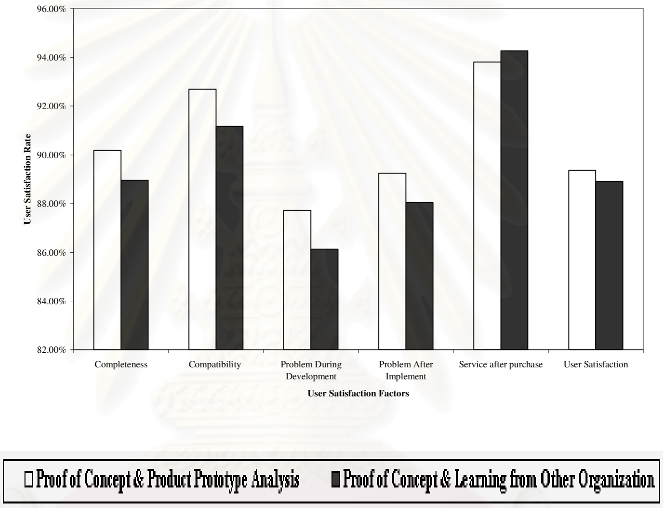


Figure 5: User Satisfaction Rate on the combination of risk management processes.

As a result from Figure 5, the combination of POC and product prototype analysis shows highest user satisfactions comparing with the combinations among POC and other risk management procedures.

5. DISCUSSION

The advantages of requirement analysis are

Requirement analysis could extract organizational demand precisely.

There are various tools that support for helping organization extract requirement such as use case, and data flow diagrams.

The overall process has ability to handle dynamically requirement.

The weakest points of standard procedure are

Extracting user requirement by using tool with inexperience officers could produce greatly error.

Since there are various tools for extracted user requirement expertise should select the appropriate tools that suite for organization culture and process.

Requirement analysis is focusing only on the requirement of organization. There is no process to analyze purchased software that can use correctly.

POC can track purchasing software information by defining succession criteria and test software features and functions based on assumption criteria. This process could help organization on filtering malfunction software. By the way, this method could not handle flexibility or dynamically requirements and take too much processing time.

Product prototype analysis could understand software nature, feature and function, interface, user friendly, and compatibility. This procedure could not handle on user requirements in detail.

As a result shows above, the trend of the user appreciation rate was significantly increasing using combination of risk management processes. Each process can be used concurrently during software selection process and promoted each other to increasingly acceptance level while standard risk management presents moderately user perception rate. This method performs better result than other risk management methods, excluding the POC and product prototype analysis.

6. CONCLUSION

Currently, Thai’s companies are relied on software to manage their business. Most software come from purchasing procedures. Unfortunately, the standard risk management method on software purchasing can not served users’ requirements. Thus, the software purchasing process becomes critical issues because its results affect the entire operations of the organizations. Therefore, Thai’s organizations have to find the better procedures to be applied on the software purchasing process.

There are several risk management procedures could be applied for handle software purchasing, including POC, product prototype analysis, software presentation, software demonstration, and learning from other organizations. As the results, the standard risk management procedure shows the moderate user satisfaction and presents the user perception is lower than the POC, product prototype analysis, and both combinations. The previous results analyze that using only one risk management process could help organization to selected software but it still be not capable. In order to obtain accuracy on software purchasing process, organizations should use various risk management process to select the appropriate software.

7. REFERENCES

[1] J.E. Bailey, and S.W. Pearson, "Development of a Tool for Measuring and Analyzing Computer User Satisfaction," *Management Science*, Vol. 29, No. 5, May 1983 pp 530-545.

[2] B. Ives, M.H. Olson and J.J. Baroudi, "The Measurement of User Information Satisfaction," *Communication of the ACM*, Vol. 26, No. 10, October 1983 pp 785-793.

[3] N.P. Melone, "Theoretical Assessment of the User-Satisfaction Construct in Information Systems Research," *Management Science*, Vol. 36 Issue 1, Jan 1990, pp. 76-91.

[4] R. Rudd. “Should Your Software Selection Process Have a Proof of Concept? Part One : Structures and the Selection Process” July 2004

[5] R. Rudd. “Should Your Software Selection Process Have a Proof of Concept? Part Two : Advantages, Disadvantages and Conclusion” July 2004.

[6] A.O. Adebowale, “Systems Development Requirements Analysis Process” *Computer Resource Center Federal University of Technology*, Akure July 2008.

[7] G.T. Clay, and P.G. Smith, "Rapid Prototyping Accelerates the Design Process." *Machine Design.* March 9,2000.

[8] T. Edmark, "Model Approach: Creating a Prototype that will Wow Investors." *Entrepreneur.* October 1997.

[9] M. Schrage, "How Prototypes Can Change Your Business." *Across the Board.* January 2000.

[10] “A Risk Management Standard” *AIRMIC, ALARM, IRM*: 2002

VITA

NAME:	Mr. Setaphat Yenyuak
BIRTHDATE:	Apr 1, 1980
BIRTHPLACE:	Petchburi, Thailand

EDUCATION:

M.Sc. in Computer Science and Information, Department of Mathematics, Chulalongkorn University, Bangkok, Thailand.
(November 2006 - March 2009)

B.Sc. in Computer Science, Department of Mathematics, Chulalongkorn University, Bangkok, Thailand.
(May 1998 - March 2002)

WORK EXPERIENCE:

Apr 2002 - Current:	CBS – RM Bangkok Bank Co., Ltd.

ACADEMIC PUBLICATION:

Yenyuak, S. and Bhattaragosol, P, "Risk Management in Software Purchasing :An Empirical Study of Thai Companies" Proceedings of the 7th International Conference on e-Business (INCEB), Bangkok, Thailand, pp. 109-114, 6-7 November 2008.

CURRENT WORK POSITION AND OFFICE:

Core Banking System – Relationship Management, Senior System Analyst

Bangkok Bank Company Limited, Technology Division
2222 Rama III Road, Chongnonsi, Yannawa, Bangkok, 10120