INVESTIGATING THE STATE OF REQUIREMENTS ELICITATION IN THAILAND

By

Sumanunya Chaiyabute

A thesis submitted in fulfillment of the requirements for the degree of Master of Engineering

Faculty of Engineering
University of Technology, Sydney
2009

Research supervisor: Mr. Ravindra Bagia

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by the undersigned. Any help that I received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Production Note:
Signature removed prior to publication.

Sumanunya Chaiyabute

ACKNOWLEDGEMENTS

Acknowledgements are given to all people for their assistance in the case studies and survey. Their enthusiasm and assistance in this research are highly appreciated.

I would like to express my sincere appreciation to my supervisor, Mr. Ravindra Bagia for providing me all invaluable kind help, suggestions and encouragement during the study that enabled me to successfully complete this study and Dr. Narelle Smith for the kind help and guidance in questionnaire revision. Without their support and recommendation, I may not able to carry out this study. Lastly, I am deeply thankful to my father, Suthep Chaiyabute and my mother, Maryat Chaiyabute for eternally love, support, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This research is aimed at investigating the state of requirements elicitation in Thailand in order to increase the understanding of the area of requirements elicitation in the country. This research is based on Thailand where the culture is different from Western countries. This relates to the previous research of Thanasankit (2002) that the hierarchical culture of Thai influences the decision making process and requirement elicitation. This hierarchical structure of Thai organization also contributes to a bureaucratic, elongated decision-making process during information systems development. Therefore, understanding the influence of power in Thai culture on requirement elicitation will assist multinational information systems consulting organizations to select, adopt, better manage, or change requirement elicitation process methodologies.

The research also identifies some of the problems encountered during the requirements elicitation process and elicitation techniques used in Thailand. Finally the results from this research are compared with best practice reported in literature review in order to provide useful suggestions and future research in this area.

To achieve the aim of the research, a questionnaire survey was employed. The researcher chose 30 construction firms and 30 software development firms in Thailand to provide the data for analysis. The firms were medium- and large-sized firms. The data from these firms were analysed with Statistical Software Packages for the Social Sciences (SPSS) version 12.

The results of the survey in this research showed:

- Software development firms were more likely to be involved with projects related to system and software development than construction firms, whose main work involved with supervising the building work.
- Construction firms seemed to have more problems or difficulties than software development firms while software development firms had better performance in requirement elicitation than did the construction firms.

- The most important source of requirements identified by both software development and construction firms was the same, namely customers or users' needs. The second important source of knowledge for both types of firms was also the same, namely domain information.
- The problem most often confronted with by both software development and construction firms was vague requirements.
- Requirement elicitation techniques information from both software development and construction firms implied that the current requirements elicitation techniques might not be appropriate. Engineers, managers and supervisors of the firms did not indicate awareness of whether the requirements elicitation techniques they used were dated or not.

The recommendations from this research are:

- Software development and construction firms in Thailand need to be more proactive and improve all stakeholders' requirement elicitation.
- Both types of firms should take the problem of vague requirements into consideration seriously when they carry out any system development. Requirement analysis techniques should be employed to facilitate this.
- It is suggested the efficiency and effectiveness of the techniques Engineers, managers and supervisors of the firms currently use for requirement elicitation be evaluated and analyzed for improvement in the RE process.

CHAPTER 1
INTRODUCTION

According to Carr (2002), the modern industrial or commercial facility is a complex high technology system made up of multiple interacting subsystems. A variety of business systems, information system, instrumentation systems, infrastructure systems, and management systems interact to make the facility work. Many companies are now involved in projects to develop complex systems. Moreover, even small organizations now develop their own systems or subcontract to develop parts of other organizations' larger systems (Carr, 2002).

Such system development process has been significantly affected by a rapid development of technologies. In the past, system designers were able to understand and predict the components that would be available when the system was finally fielded or installed at customer sites. Follow-on support was less complicated than it is today because component parts were available for decades (Carr, 2002). However, systems today which use digital processors or computers require rapid and flexible development process since the technology emerges, lives its life-cycle, and fades away into oblivion within the short period of time (Carr, 2002). This necessitates a merging between management and engineering functions. Technical management is essentially an engineering discipline and is often carried out within Systems Engineering.

One of the key areas that system engineering emphasizes is "Requirements Engineering". It does so because of the recognition that unless the requirements for a system or project are correctly captured and managed, the system created by the project may well fail to meet its "fitness for purpose" criteria (Easterbrook, 2004). There is ample evidence in the literature regarding projects that have failed due to requirement problems. Some of these are considered below.

Thailand is now getting involved with complex projects involving complex systems and software. There are a number of examples of failed projects emerging from Thailand which seem to be a result of poor requirements engineering. Yet there is not much research reported in the literature about the state of requirements engineering in

Thailand compared with best practice. This research aims to investigate a specific element of requirements engineering practice in Thailand, with a view to contributing towards improvement in that practice. According to Carr (2000), requirements are the descriptions of properties, attributes, services, functions, and/or behaviors needed in a product to accomplish the goals and purposes of the system:

"If it mandates that something must be accomplished, transformed, produced or provided, it is a requirement" (Harwell et al., 1993)."

The specific element looked at is the elicitation of requirements from the stakeholders.

1.1 Problems in requirements elicitation

Poorly done requirements engineering can result in serious project failures: late delivery, over budget, and either poor or incorrect performance (Carr, 2000). As can be seen from Figure 1.1 (the relationship between costs of fixing a problem and the phase in which it was discovered), the later the problem is fixed, the higher the cost involved, with dramatic escalation of costs occurring after fielding (Carr, 2000). GAO (GAO, 1992) also reports that problems in the requirements are major causes of failure in two-thirds of the system. Requirements were reported to be the most significant problem in developing software systems in a number of companies (Faulk et al., 1992). Problems related to incompletion and inconsistence of requirements at the initial development phase of the software is one of the main obstacles of the software development process (Santander and Silva, 2007). This reflects the importance of the initial step of requirement captures (or requirement elicitation[1]).

---

[1] "Elicitation" and "Capture" are not exactly the same. "Elicitation" is preferred to "Capture" to avoid the suggestion that requirements are out there to be collected simply by asking the right questions (Goguen and Jirotka, 1994).

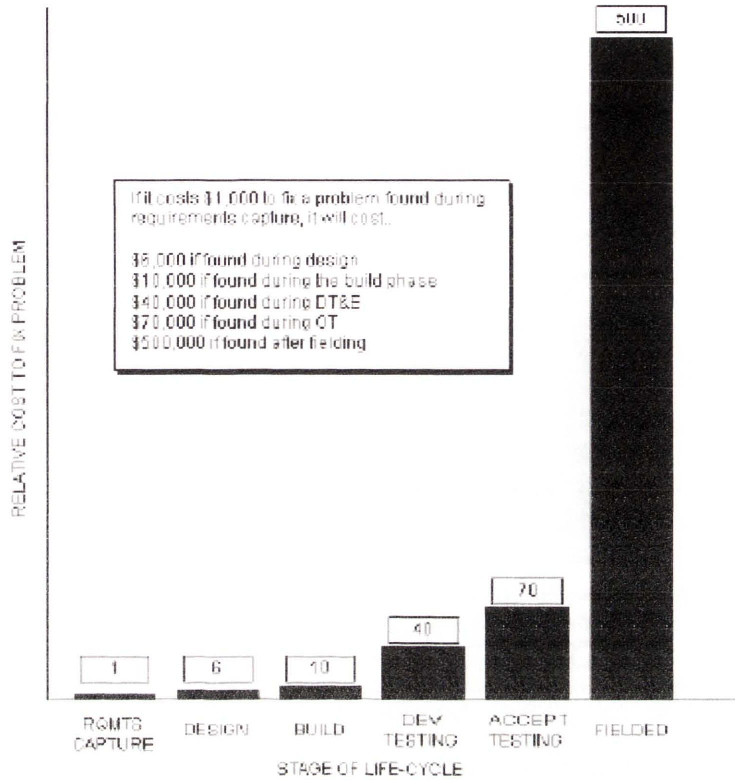

Figure 1.1: Cost ratio of fixing a problem in systems design

(Baseline: requirement capture = 1) (Carr, 2000)

In the process of requirements elicitation, Carr (2000) has reported a number of problems:

- requirements do not reflect the actual needs of the system stakeholders
- inconsistent requirements; incomplete requirements
- conflicting requirements; misunderstood requirements
- ambiguous or vague requirements
- introduced requirements
- spurious requirements
- unintended consequences

Nuseibeh and Easterbrook (2007) have also indicated stakeholders (including paying customers, users and developers) may be numerous and distributed. Their goals may vary and conflict, depending on their perspectives of the environment in which they work and the tasks they wish to accomplish. Moreover their goals may not be explicit or may be difficult to articulate, and, inevitably, satisfaction of these goals may be constrained by a variety of factors outside their control. These cause problems at requirement elicitation process. In the context of Thailand, Thanasankit and Corbitt (2000) have reported that elicitation is affected by use of different analytical and elicitation tools. Problems such as an inability to gain full access to all users where requirements can be identified; increased time needed to uncover requirements; and inability to develop requirement specification, were also reported. Therefore this research is aimed as investigating the state of requirements elicitation in Thailand.

1.2 Examples of projects failures and project successes

As mentioned above, many projects failed due to poor requirement engineering, and errors in requirements engineering can propagate through the entire development process. Therefore, if we focus on any problems that may occur in requirement elicitation and address them, we can reduce the risk of serious problems such as delay, time and money waste, and project failure. The following are some case studies that show how important requirements are. These projects failed because of poor requirement understanding.

Edsel

In the past, Ford automobile company produced only "black" cars that were very popular at that time. However, Ford tried to improve the cars with a new look by introducing "Ford Edsel". Unfortunately, the Ford Edsel was commercially unsuccessful because people did not want it. They did not like the cars because they were too big, had too many functions and were too costly. These cars were inappropriate as they were not what the customer wanted. (Kumarnbun, 2007)

IBM PCjr

This is one of the more renowned failures. IBM created an inexpensive computer for the classroom. However, they were substandard compared with non IBM-PC of the time and their keyboard was too small for normal sized fingers. Therefore the people did not buy them. This shows again a failure to appropriate user needs. One lesson that has been learnt is reflected in normal sized keyboards on laptops, and styluses for PDAs. (Bahill and Henderson, 2004)

New Coke

In 1985, Coca-Cola Company changed its Coke formula to make it taste like Pepsi because they thought most of people preferred Pepsi. They released this as New Coke. However, the response was overwhelmingly negative because people did not like New Coke and did not buy it. Finally, the original formula was brought back on the market within three months. This shows that Coca-Cola Company failed to correctly consider the user requirement. (Bahill and Henderson, 2004) The literature on Quality Improvement is full of similar examples and guidance against such moves.

Breeze Tablet

"Breeze tablet" is a tablet-sized washing detergent in a small package. Unilever Company created breeze tablet with the idea of convenient detergent which users simply throw into the washing machine and then washing. Unfortunately, customers preferred to use powder detergent rather than tablets, since they wanted to control the amount of powder detergent to use in loads. (Mongkolpong, 2008)

Ramintra - at Narong Expressway Construction Project, Thailand

This expressway was constructed in order to reduce the traffic population in the city area. However, this project is not successful because people do not use the expressway. This expressway was built overlapping with another way which belongs to a different company (Japan Bank for International Cooperation (JBIC), 2004). Also, people have to pay a threefold fee three times to go through. Consequently, they prefer using the overlapped road, which is free. It can be seen that the company failed to consider that people will not use an expressway they have to pay for while they have a free alternative. (JBIC, 2004)

Trans-Thai-Malaysia Gas Pipeline Project

Trans-Thai-Malaysia Gas Pipeline project belonged to Petroleum Authority of Thailand (PTT). Problems in this project are due to lack of people involvement until project planning was almost completed. Even though many consultant companies were employed, they did not do the research with local people. This project had to be stopped as people did not accept it because of environmental and other reasons. This project failed to consider environment conservation, which the stakeholder considered as very important requirements. Therefore, doing research with local people at the initial stage of project was essential. (Phasukvanich, 2003)

Mae Mo Power Plant

Mae Mo power plant is located in Lampang, Thailand. This project was designed to provide nearly 3,000 megawatt of power; however, in fact only one forth of overall power was produced due to the need to prevent air pollution. This shows that another three-fourths of the investment was wasted. However, Electricity Generating Authority of Thailand (EGAT) has subsequently provided sulfur dioxide removal equipment, and reduced the mining affects through discussions with people with domain knowledge. This has allowed the power from Mae Mo power plant to be increased up to 1300 megawatt. Nevertheless, this is still 50% less than its full capacity. This shows requirements in this project were not well developed, and that EGAT failed to consider the environmental issues. (Phasukvanich , 2003)

However, in contrast, if companies pay more attention to user requirements and other constraints then they achieve their goals and win the market. The following case studies are examples of Thai companies that appear to have done so.

Black Canyon

Kotler, Kartajaya, and Huan (2007) wrote in "*Think ASEAN! Rethinking Marketing toward ASEAN Community 2015*" that Black Canyon, Thai coffee shop, is successful because they care about customers. Black canyon does research about customers in every area because they are different. For example, in Dubai some customers do not eat pork, then Black Canyon decided to add special menu such as seafood and serve them more food in every dish since they consume double compared with Asian people. Also, shop decoration in Dubai is high class because most of them are rich people and they prefer premium quality. These exactly meet the customer requirements that made Black Canyon win today.

Mama Instant noodle

Paniangvait (2007) states that Thai President Foods Plc provided Mama instant noodle "Tom Yum Goong" flavor which is spicy. This is because they found that Thai people love hot and spicy taste and no one had created this taste for instant noodle before. For that reason, they are still unbeaten because they focused on what customers desired.

7-11 Mini-marts

7-11 is the well known mini-mart. Yongpisarnpob (2007) informs that she manages several 7-11 shops in Thailand. Each shop would be managed in different ways in different areas. For example, at Mochit bus terminal, a person who is sitting and waiting for buses does not want to loose their chair. Therefore, staff at 7-11 Mochit decided to carry some snacks, foods, and drinks for selling there. This convenience was very popular with the customers. Also, on Songkran festival at Khao San Road, 7-11 are typically overflowing with people, so they remove the shelves in the middle of shop. Moreover, some oils, fish sauce, sugar, and some unpopular product were temporary removed as well. This is because customers prefer to buy other snacks and drinks. Moreover, they also provide what tourists need at that time, for instance, sandals and some powders.

Such examples illustrate the success that may be achieved through proper focus on requirements and needs.

1.3 Requirements engineering research and requirements elicitation in Thailand

It can be said that requirements engineering research in Thailand is not as intensive as in other developed or Western countries (Asanghanwa et al., 2005). Very few studies about requirements engineering in Thailand have been found in the literature. The following discusses some that have been found.

Firstly Thanasankit and Corbitt (2000) have studied cultural context and its impact on requirements elicitation in Thailand. An ethnographic study of how software analysts in Thai software houses undertook the requirements engineering process was carried out. Comparison in the nature of elicitation of requirements by system analysts in Thailand with standard descriptions of elicitation in requirements engineering research was also made. The study employed Hofstede's (1991) framework to interpret the impact of Thai culture.

The results showed that many multi-national corporations, which were currently operating in Thailand, were facing difficulties in dealing with their clients. Misinterpretation of their client's requirements and behaviour led to misunderstanding their clients and sometimes led to cancellation of projects. This problem might also be caused by applying western ways of thinking and analysing of situations.

The results suggested that each country needs its own distinct methodologies for collecting requirements and an Information System Development Methodology (ISDM) (Korpela et al., 1998). Understanding local culture helps systems analysts in adapting current methodologies for elicitation to suit each country. Moreover understanding local culture may inevitably improve the quality outcome of the systems development process. Holmes and Tangtongtavy (1995) have also suggested that while addressing multi-national practices of information system strategy, system analysts should consider the use of such practices since they might be affected by local culture.

The Thanasankit and Corbitt (2000)' research also suggested that understanding culture would assist system analysts in understanding their client's work practices and understanding certain behaviour, which were shared between people in a particular society. Understanding Thai culture and its implication toward requirements engineering influenced successful design and acceptance of new information systems. These suggestions could be applied to gathering requirements for development of information systems in Thailand. Thai culture is still dominant in Thailand everyday life. International Information Systems Consulting organizations who want to operate in Thailand need to place their first priorities on understanding Thai culture because it goes deeper than a simple list of local etiquette (Holmes and Tangtongtavy, 1995).

Based on Hofstede's (1991) framework, Thanasankit and Corbitt (2000) have reported, through interviews, the difficulty that Thais felt in decision making. Thai society constructs its reality in accordance with group or social interests rather than individual interests. Trust and relationships with others are the basis of the Thai culture, supporting Sorod's (1991) view that Relationship-oriented behaviour happens more commonly than work-orientated behaviour in Thai society and its organizations (Gray and Campbell, 2002). They further supported Jirachiefpattana's (1996) suggestion that personal and family connections play an integral part in operations of Thai business, where views and opinions have a greater impact on business management when expressed by members of family or in-group members (Gray and Campbell, 2002). Thanasankit and Corbitt (2000) research has further suggested that power in the Thai context is constructed not by influence or personality, rhetoric or education, but rather through position and the status associated with position and rank. Since "trust and relationships with others are the basis of the Thai culture, relationship oriented behavior happens more commonly than work-orientated in Thai society and its organizations" (Gray and Campbell, 2002). Decision-making process of Thai people is often not based on a team approach, unlike Western decision making. Subordinates in Thai organizations accept that their superiors make decisions in an authoritarian way (Thanasankit and Corbitt, 2000), and Thai culture does not encourage subordinates to dare to make mistakes, or to take initiatives (Gray and Campbell, 2002).

Thanasankit (2002) explores the impact of Thai social status and hierarchical decision-making processes during Requirements Engineering (RE) in business information systems development projects. The paper illustrates the interaction of RE processes and national culture, and shows that culture can change the way processes are implemented, rather than process changing culture. Thai culture is naturally inherent in Thai daily life and Thais bring that into their work information practices. The concepts of power in Thai culture contributes toward hierarchical forms of communication and hierarchical relationships between systems analysts and decision-making process in Thai software houses, especially during RE, where information systems (IS) requirements need to be established for further development. The research shows that power significantly influences the decision making process and the construction of social status between clients and systems analysts in Thailand. The hierarchical structure of Thai organizations also contributes to a bureaucratic, elongated decision-making process during information systems development (ISD). Understanding the influence of power in Thai culture on RE will assist multinational IS consulting organizations to select, adapt, better manage, or change RE process methodologies.

1.4 Research Justification

The main reasons for choosing "Investigating the state of requirements elicitation in Thailand" as a research topic are the followings:

Firstly, there are very few studies on requirements engineering and requirements elicitation in Thailand. According to Asanghanwa et al. (2005), there is likely a relationship between requirements engineering research and economic prosperity. Increasing RE research might have a positive impact on the economic health of a country. This research is expected to make a contribution to improvement in the RE practice in the country.

An understanding of the state of requirements elicitation practice in Thailand and the problems and difficulties the system analysts have encountered can lead to a development of a more holistic understanding of the requirements process. As a result, companies can more effectively organise and manage requirements, i.e., in terms of cost reduction, a decrease in numbers of failures and the delivery time.

Secondly, requirements are reported to be the most significant problem in developing software systems in a number of companies (Faulk et al., 1992). Problems related to incompletion, and inconsistency of requirements at the initial development phase of the software is one of the main obstacles of the software development process (Santander and Silva, 2007). Moreover for almost 10 years there have been attempts in Thailand to improve software quality (Pongpaibul and Boehm, 2005). A study in requirements elicitation and requirement engineering is expected to be indicate to some extent, the effectiveness of these efforts to highlight any interdependencies between problems encountered and requirement elicitation practice.

Thirdly several problems involving requirements engineering and requirements elicitation have been reported in the literature. Thanasankit and Corbitt (2000) have reported that requirements elicitation in Thailand is affected by:

- Different use of analytical and elicitation tools
- Different adaptation of the analytical and elicitation tools to the elicitation process
- An inability to gain full access to all users where requirements can be identified
- Increased time needed to uncover requirements
- An inability too develop requirements specification

Appropriate solutions can only be envisaged for problems if the latter are clearly and precisely identified. Therefore it will be beneficial to determine problems associated with requirements elicitation in Thailand.

1.5 Aims and Objectives

1.5.1 Research questions

These are some questions that have been used as a guide to formulating the objectives of this research. The questions include:

- What is the current state of requirements elicitation practice in Thailand?
- How effective is it?
- What are the main problems that occur during the requirement elicitation process?
- Why do they occur?
- How may they be addressed or solved?
- How does the Thai practice compare with best practice and with theory?

1.5.2 Main Objective

The main objective of this research is to investigate the state of requirements elicitation practice in Thailand. However, though the main focus is on requirement elicitation, the research also considers requirement constraints and compliance to requirements in a system.

1.5.3 Sub-objectives

The following sub-objectives are included:

- To increase the understanding of the area of requirements elicitation in Thailand
- To determine the problems encountered during the requirements elicitation process in Thailand
- To evaluate elicitation techniques used for requirements elicitation in Thailand
- To compare the results with best practice and theory in order to provide useful suggestions for practice and future research in the area of requirements elicitation.

1.6 Benefits

- *Organizations:* understanding the state of requirements elicitation in Thailand and the problems and difficulties the system analysts have encountered can lead to a development of a more holistic understanding of the requirements process. Consequently, companies can more effectively organize and manage requirements.

- *Software System:* requirements are reported to be the most significant problem in developing software systems in a number of companies (Faulk et al., 1992). Therefore a study in requirements elicitation and requirement engineering is expected to be one of the key factors for improving software quality in Thailand.

- *Requirement Engineer:* This research will help the engineer to understand the gaps between their practice and best practice, and to address these gaps in as appropriate manner.

1.7 Research Methodology

Research methodology details are discussed in Chapter 3. Here we summarise some key aspects only.

Secondary and primary data were used for this research. Secondary data was obtained from reviewing the related literature from several kinds of sources. These include books, academic journals, and electronic data bases.

Primary research in this study focuses on quantitative (questionnaires) approach. Self-completed questionnaires were employed. Samples for this research were drawn from people from companies who work in software and construction companies in Thailand and have responsibility in requirements elicitation (e.g. software engineers, system developers, business systems analysts). They were selected on the basis of the following broad criteria:

- Companies should be large enough to be involved with projects requiring use of requirement elicitation processes.
- Companies would likely use a formalized process of project management and requirements engineering.
- Companies should be chosen from the domain of software engineering and construction, so that an understanding of state of Requirement Engineering in these domains may be achieved.

120 companies will be targeted for this research. Assuming a 50% response rate, it is anticipated that the sample size of around 60 will be obtained.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter provides a critical review of the literature pertaining to the research objectives. It discusses aspects of Systems Engineering first to establish a context for Requirements Engineering and, in particular, Requirements Elicitation. Definitional aspects of requirements are considered, as are some classifications of them, together with some of the common techniques and problems of requirements engineering and elicitation.

2.2 Systems engineering

Systems engineering is a complex process that involves requirements definition and analysis, design, coding, testing and implementation (Marasco, 2006). A "System" is typically defined as an interacting combination of elements to accomplish a defined objective; these elements include hardware, software, firmware, people, information, techniques, facilities, services, and other support elements. (International Council on Systems Engineering, INCOSE, 2000 cited in Carr, 2000).

2.2.1 Definition of system engineering

System engineering is defined as an interdisciplinary approach to evolving, verifying and managing a balanced, integrated set of system product and process solutions that satisfy customer needs over the life-cycle of the system (INCOSE 2000 cited in Carr, 2000). It is that part of the technical management process that coordinates and oversees the translation of an operational need into a system designed to meet those needs (Johns Hopkins University, 1998).

System engineering evolves several methodologies for developing and fielding systems. There are several approaches used for system development. A traditional approach is called "Waterfall model" as shown in Figure 2.1. According to this model, each step is discrete and needs to be completed before the following step. As can be seen from the figure, *requirement capture* is the first step of system development. During this period the systems designer will try to evaluate what the customer wants, and then prioritize those statements. This step is in the area of requirements definition wherein the greatest potential for success or failure exists (Carr, 2000). Requirements gathering and specification are accomplished by the process of requirements engineering (Austin, 2007). Requirements engineering (RE) is treated as a phase of systems engineering whose ultimate goal is to deliver some systems behavior to its stakeholders (Stevens et al., 1998) or a branch of software engineering concerned with the real-world goals for functions of and constraints on software system (Zave, 1997).

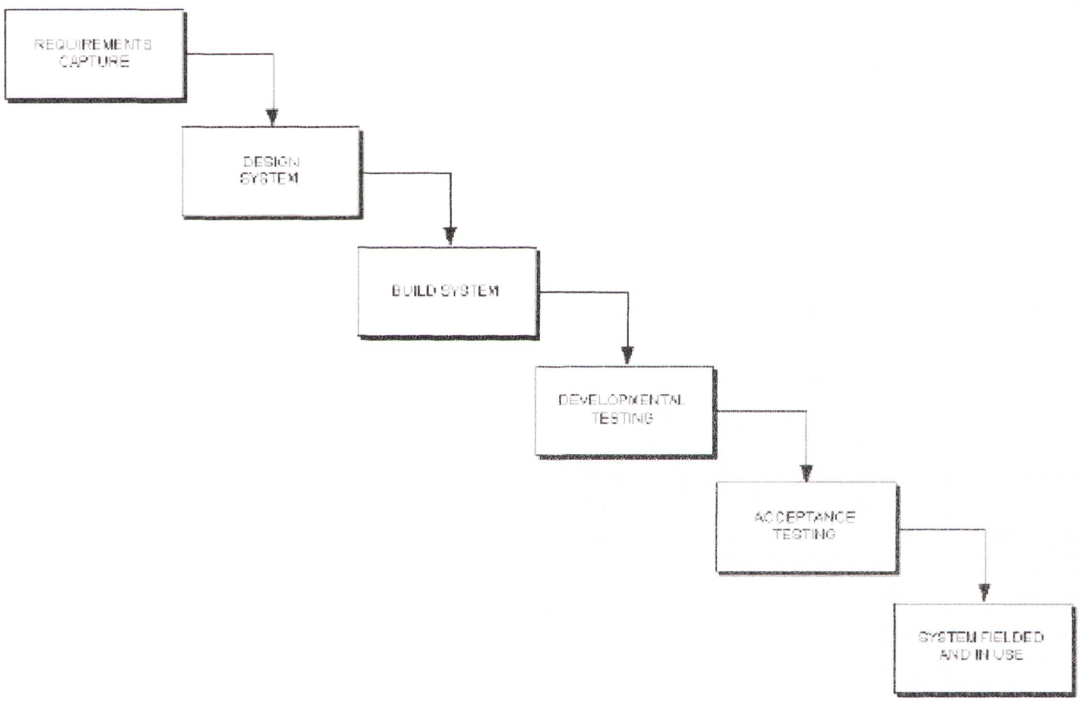

Figure 2.1: Modified "Waterfall" model of system development (Carr, 2000)

2.3 Requirements engineering

2.3.1 Definitions of requirements engineering

Several definitions of requirements engineering have been given in the literature. In relation to software engineering, the following definitions are given.

Zave (1997: 315) has defined requirements engineering in the context of software development as "the branch of software engineering concerned with the real-world goals for functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families".

Similarly, software systems requirements engineering (RE) is broadly defined as the process of discovering the purpose for which it was intended, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation (Nuseibeh and Easterbrook, 2007). It is defined as all the activities devoted to identification of user requirements, analysis of the requirements to drive additional requirements, documentation of the requirements as a specification, and validation of the documented requirements against the actual user needs (Saiediana and Dale, 2000).

Asanghanwa et al. (2005) have defined requirements engineering as the discipline of determining, analyzing, pruning, documenting, and validating the needs and requirements of stakeholders for a software system. Research in requirements engineering has a main purpose to improve society's knowledge and practice of the discipline so as to positively impact society's ability to build new, ever more effective software systems.

Sommerville and Sawyer (1997) have stated that requirements engineering covers all of the activities of discovering, documenting, and maintaining a set of requirements for building a computer-based information system. Requirements engineers perform requirements analysis and definition on specific projects as well as investigate in the abstract how requirements should be developed (Dorfman, 1999).

These definitions seem to focus on software engineering. In reality software cannot function in isolation from the system in which it is embedded, and hence requirements engineering has to encompass a system level view (Nuseibeh and Easterbrook, 2007). Therefore the following definitions of requirements engineering are more related to system development and system engineering discipline as following.

Requirements engineering focuses on improvements to the front-end of the system development life-cycle by establishing the needs that have given rise to the development process and organizing this information in a form that will support system conception and implementation (Finkelstein, 1994).

According to Marasco (2006), requirements engineering is a phase in system development that entails all the activities of user requirements identification, analysis of the so collected requirements to identify more requirements, converting the requirements into software systems requirements specification document (SRS document) and then validating and verifying the SRS document in relation to the user needs.

Loucopoulos and Karakostas (1995: 13) have defined requirements engineering as "the systematic process of developing requirements through an interactive co-operative process of analyzing the problem, documenting the resulting observations in a variety of representation formats, and checking the accuracy of the understanding gained."

Arayici et al. (2005) have reported that requirements engineering is concerned with the goals, desired properties and constraints of complex systems that involve software systems, organizations and people. It covers all activities related to the acquisition, specification and maintenance of requirements throughout the life cycle. It also includes how requirements relate to business processes, soft issues, work redesign, system and software architecture and testing.

2.3.2 Importance of requirements and requirements engineering

According to Thanasankit and Corbitt (2000), the important role of requirements for software and systems development can be traced to earlier studies of software engineering. Errors during the requirement phase have been recognized by software developers as leading to high costs in fixing systems and, in the worst case, the rejection of system (Laudon and Laudon, 1995). The cost of correcting an information system mainly comes from poor design where system analysts could not clearly understand the requirements of the new or upgraded information system (August, 1991; Thanasankit and Corbitt, 2000). 40% of the cost of an information system were generated by error removal, of which 45-65% of the errors occurred during systems design (Rush, 1985). Martin (1984) has also reported that 64% of the errors occurred during analysis and design phase although the design had been formally approved by users. These errors were generated from improperly described system or improperly designed system (Martin, 1988). It was suggested by Chatzoglou et al. (1996) that one-third of information system projects were completed without capturing all necessary requirements. The later the system analysts discover their errors during the system development life cycle (SDLC), the more it costs to fix (Boehm, 1981). For these reasons, the early stage of SDLC, which is the requirement phase, has been received significant focus.

Dorfman (1999) has stated that the benefits of good requirements include:

- Agreement among developers, customers, and users on the job to be done and the acceptance criteria for the delivered system
- A sound basis for resource estimation (cost, personnel quantity and skills, equipment, and time)
- Improved system usability, maintainability, and other quality attributes
- The achievement of goals with minimum resources (less rework, fewer omissions and misunderstandings)

In addition, the benefits of good requirements and the criticality of doing them well increase considerably with the size and complexity of the system being developed. Additionally, software-intensive systems seem to have more inherent complexity, that is, are more difficult to understand, than systems that do not contain a great deal of software; thus these systems are more sensitive to the quality of their requirements (Dorfman, 1999).

Requirements engineering is one of the most crucial steps in software development process (Saiedian and Dale, 2000). This is because that solution to problems related to incompletion and inconsistence of requirements at the initial development phase of the software is one of the main obstacles of the software development process (Santander and Silva, 2007).

Although the relationship of RE and software system success has not been proven (Davis and Zowghi, 2005), it is certainly generally believed that more effective RE practices will increase the probability of success. For example, the Standish Group

(2005) has shown that a large percentage of software systems that have been built failed to meet the needs of the intended users. Jones (1991) has published data showing that a very large percent of project failures could be blamed on poorly understood requirements. Since the primary purpose of RE is to increase all parties' level of understanding of requirements, it seems reasonable to conclude that improving the practice of RE will increase the number of software system project successes (Asanghanwa et al., 2005). Moreover, since software systems now have been involved in almost every aspect of society, success in developing such systems should provide a significant benefit to the economies of countries utilizing them.

Research in requirements engineering is important because it is thought as the 'front end' of the system life cycle (NATURE Team, 1996). The high cost of system errors which originate during requirements analysis has been acknowledged and many system failures have been attributed to mistakes in the early phases of system development (Bell and Thayer, 1976 cited in NATURE Team, 1996). Poorly done requirements engineering can cause serious project failures: late delivery, over budget, and either poor or incorrect performance (Carr, 2002). In addition, the subject of requirements engineering is inherently broad, interdisciplinary, and open-ended (Zave, 1997). Therefore it is crucial to carry out research in the field of requirements engineering, and hence the focus of this research on requirements elicitation.

2.3.3 Definitions of requirements

According to Carr (2000), requirements are the descriptions of properties, attributes, services, functions, and/or behaviors needed in a product to accomplish the goals and purposes of the system. In other words:

> *"If it mandates that something must be accomplished, transformed, produced or provided, it is a requirement" (Harwell et al., 1993)*

According to Abran et al. (2005), system requirements are the requirements for the system as a whole, and in a system containing software components, software requirements are derived from system requirements.

2.3.4 Types of requirements

Several classifications of requirements are reported in the literature.

Abran et al. (2005) have divided requirements into product requirements and process requirements. Product requirements are requirements on product to be developed while process requirements are essentially process constraints on the development of the system. These two types however are not completely distinct. Some product requirements generate implicit process requirements, for example, the choice of verification technique. Another might be the use of particularly rigorous analysis techniques (such as formal specification methods) to reduce faults which can lead to inadequate reliability. Process requirements may also be imposed directly by the development organization, their customer, or a third party such as a safety regulator (Kotonya and Sommerville, 2000; Sommerville and Sawyer, 1997).

Requirements are also classified in general as user requirements and system requirements (also called functional requirements). User requirements are statements in simple natural language of users needs from the user perspective and in terminologies best understood by the user. They describe tasks the user will be able to perform with the system (Charvat, 2003). System requirements are detailed descriptions in technical terms of what the proposed system will do. They also describe the behavior of the system or a property or an attribute of the system. Moreover system requirements encompass user requirements; requirement of other stakeholders (such as regulatory authorities), and requirements without an identifiable human sources (Abran et al., 2005).

Requirements can also be classified into functional requirements and nonfunctional requirements. Functional requirements, or sometimes known as capabilities, describe the functions that the system is to execute (i.e. formatting some text or modulating a signal), while nonfunctional requirements, or sometimes known as constraints or quality requirements, are the ones that act to constrain the solution.

Requirements can also be classified according to whether they are performance requirements, maintainability requirements, safety requirements, reliability requirements, or one of many other types of software requirements (Kotonya and Sommerville, 2000; Sommerville and Sawyer, 1997).

2.3.5 Requirements engineering framework

In the Software Engineering Body of Knowledge (SWEBOK), requirements engineering is described using a four-step process model (Figure 2.2), including requirements elicitation, analysis and negotiation, documentation, and validation (Sawyer and Kotonya, 2001). The outcome of this process is the set of agreed-upon requirements. Although this discussion uses software development to describe the RE process, the remarks apply to requirements engineering in general, except as noted specifically from time to time.

Requirements elicitation is reported as the first stage in building an understanding of the problem that the software is required to solve. Requirements analysis has to do with the process of analyzing requirements to detect and resolve conflicts among requirements, discover the bounds of the system and how it must interact with its environment, and elaborate user requirements to software requirements (Mead, 2003). Requirements negotiation involves with resolving conflicts, i.e. those that might occur between stakeholders, or between requirements and resources. Requirements validation is involved with checks for omission, conflicting requirements, and ambiguities (Mead, 2003).

In addition, Loucopolous and Karakostas (1995) have reported other framework of requirements engineering process which is an iterative process defined by direction and order and by the development and implementation of a structured process. One process can lead to or affects another in a cyclical, iterative process as shown in Figure 2.3. Nuseibeh and Easterbrook (2007) have reported 5 core activities of requirements engineering: eliciting requirements; modeling and analyzing requirements; communicating requirements; agreeing requirements; and evolving requirements.

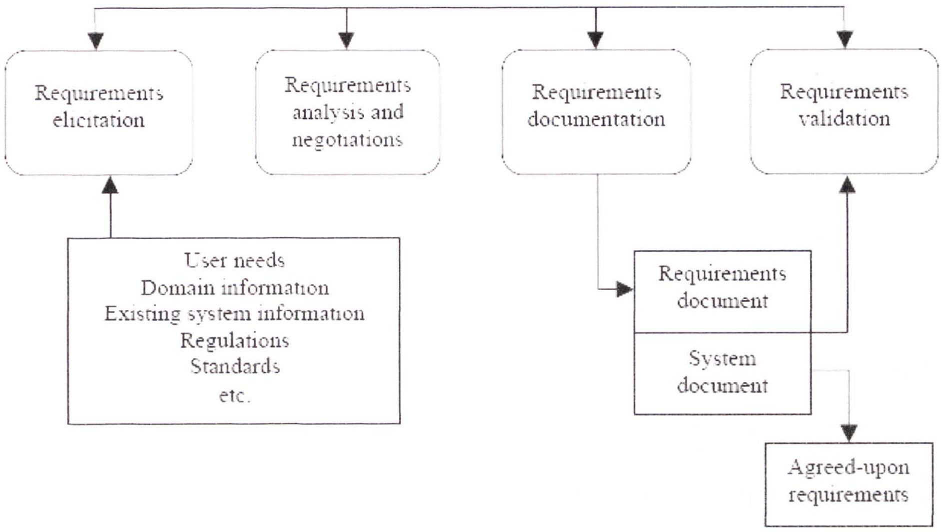

Figure 2.2: Coarse-Grain Requirements Engineering Process (Mead, 2003)

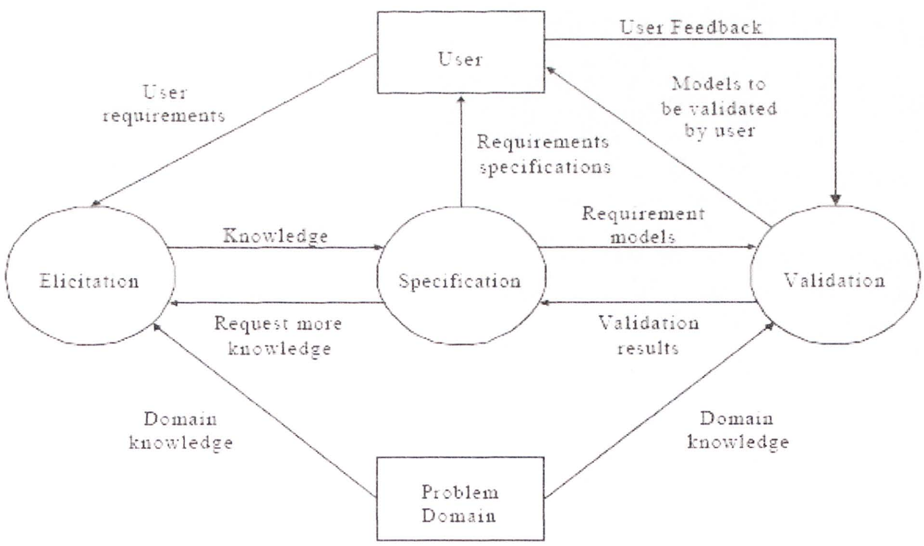

Figure 2.3: Loucopolous and Karakostas's requirements engineering framework (Loucopolous and Karakostas, 1995: 21)

Abran et al. (2005) have proposed the scope of software requirements as shown in Figure 2.4.

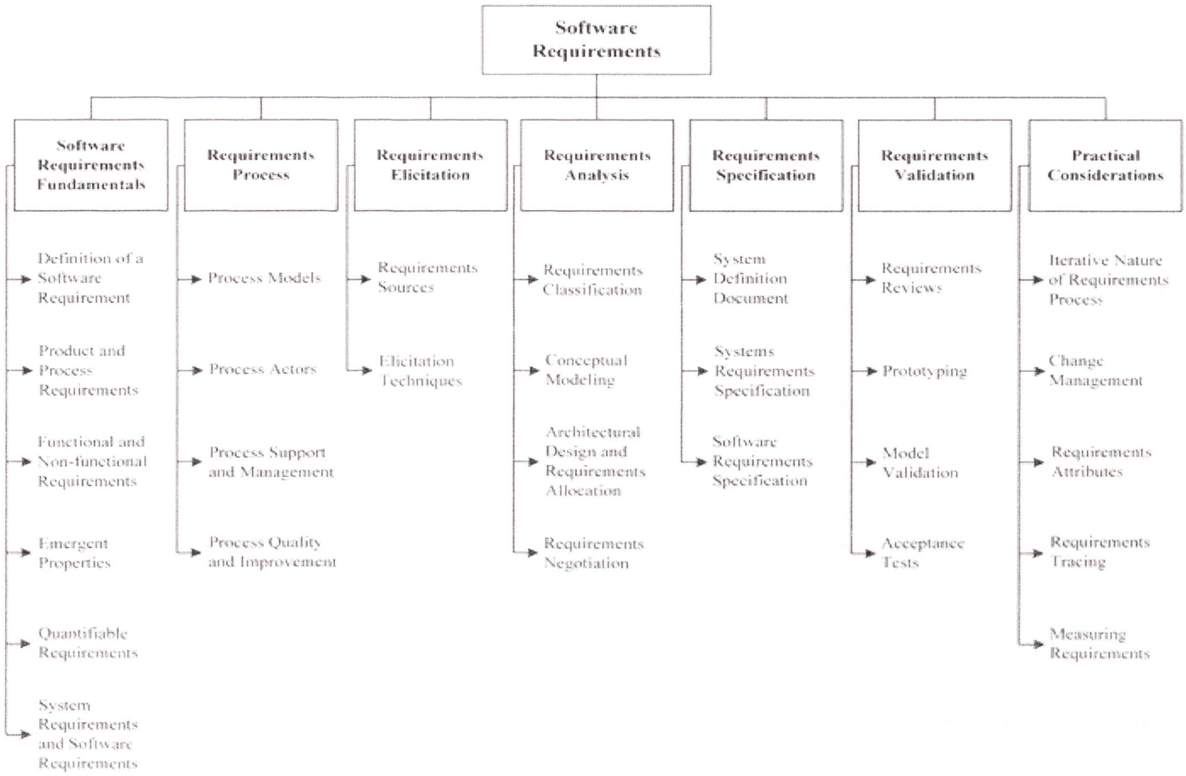

Figure 2.4: Software requirements framework (Abran et al., 2005)

2.4 Requirements elicitation

As can be seen from both Figures 2.2 and 2.3, requirements elicitation is the first step of requirements engineering process (Nuseibeh and Easterbrook, 2007). It can be considered as one of the most important step since it is closely related to other RE activities – information or knowledge gathered during requirements elicitation has to be interpreted, analyzed, modeled and validated before the requirements engineer can feel confident that a complete enough set of requirements of a system have been collected. Finkelstein (1994) has stated that the most difficult task in requirements engineering is information gathering - that is gathering information on the needs and the "domain" or "environment" in which these needs are situated. This information may exist in the form of large documents, be held by identifiable experts, be buried in the work practices of individual users, and so on (Finkelstein, 1994).

2.4.1 Definitions of requirements elicitation

Loucopolous and Karakostas (1995: 40) have defined requirements elicitation as the process of acquiring (eliciting) all the relevant knowledge needed to produce a requirements model of a problem domain.

According to Saiediana and Dale (2000), software requirements elicitation is the specific processes of gathering, determining, extracting, or exposing software requirements. From the user's perspective, good elicitation results in them having a better understanding of their needs and constraints.

2.4.2 Importance of requirements elicitation

Effective requirements elicitation provides benefits to both users and developers. Good elicitation makes users more understanding of their needs and constraints, as a result, they will be able to effectively evaluate solution alternatives and understand the implications of their decisions (Saiedian and Dale, 2000). For developers, good requirement elicitation constructs a clear, high-level specification of the problem that is to be solved. It ensures that a solution is being developed for the right problem and that the solution is feasible (Saiedian and Dale, 2000). Most importantly, a good elicitation process builds a common vision of the problem and the conceptualized software solution between the user and developer.

During software requirements development process, there are certainly many participants involved. They have different interests and play specific roles and this must be recognized in order to ensure the right people are involved at the right time and that the right expectations are addressed. Failure to do so results in indirect and ineffective communication between the customer and developer. Issues that are not addressed until later phases in system development can become major schedule and budget impacts, at best, and even potential showstoppers (Saiedian and Dale, 2000).

The study by Thanasankit and Corbitt (2000) has shown that requirements elicitation is developmental, complex, messy and chaotic rather then ordered ,structured and iterative.

2.4.3 Context of requirements elicitation

Thanasankit and Corbitt (2000) have stated that in the process of requirements elicitation, the following tasks are normally performed:

- Identifying all the sources of requirements knowledge
- Acquiring the knowledge
- Deciding in the relevance of the knowledge to the problem in hand
- Understanding the significance of the elicited knowledge and its impact on the software requirements

Abran et al. (2005) have divided the context of requirements elicitation into requirements sources and elicitation techniques.

2.4.3.1 Requirements sources

There are various sources of requirements (Figure 2.2) such as users' needs, domain information, and existing system information. Requirements can be seen to be of two types: user requirements and domain knowledge requirements (Figure 2.3). The NATURE team (1996) has stated that requirements may occur into two sub-types: user-defined and the domain-imposed requirements, but both have one end point, a requirements specification.

- *User-defined requirements* are the needs for computer systems which arise from people's requests. These are initially expressed in language and other informal means, such as diagrams, and sketches (NATURE team, 1996). According to Austin (2007), user requirements are the key to a successful product development.

- *Domain-imposed requirements are* facts of nature, including the social, organizational, and technical context, which have to be satisfied for a system to work, and form the connection between the real world and the future designed system (Jackson, 2001).

Abran et al. (2005) have identified various sources of software requirements and of the frameworks for managing them as follows:

- Goal (sometimes called "business concern" or "critical success factor"): refers to the overall, high-level objectives of the software. Goals provide the motivation for the software, but are often vaguely formulated. Software engineers need to pay particular attention to assessing the value (relative to priority) and cost of goals. A feasibility study is a relatively low-cost way of doing this (Loucopoulos and Karakostas, 1995)

- Domain knowledge: the software engineers need to acquire, or have available, knowledge about the application domain. This enables them to infer tacit knowledge that the stakeholders do not articulate, and assess the trade-offs that will be necessary between conflicting requirements (Abran et al., 2005)

- Stakeholders: much developed software is unsatisfactory because it has focused the requirements of only one group of stakeholders at the expense of those of others. Therefore, software is delivered which is difficult to use or which subverts the cultural or political structures of the customer organization (Abran et al., 2005). It is necessary for the software engineer to identify, represent, and manage the viewpoints of many different types of stakeholders (Kotonya and Sommerville, 2000). Suzanne (2000) also recommended that consults system and stakeholders of the project should be identified at the first stage of a project. The reason stakeholders is necessary is because it is much better opportunity that people will be involved to the project. Also, they are the composition of the project team which has a strong bearing on project outcomes (Coughlan et al., 2003). Therefore, knowledge and better understanding about the project would be contributed. There are several techniques can be used to identify stakeholders such as stakeholder analysis template (Sommerville and Sawyer, 1997).

- The operational environment: requirements will be derived from the environment in which the software will be executed. These may be, for example, timing constraints in real-time software or interoperability constraints in an office environment (Abran et al., 2005). These must be actively sought out, because they can greatly affect software feasibility and cost, and restrict design choices (Thayer and Dorfman, 1997).

- The organizational environment: software is often required to support a business process, the selection of which may be conditioned by the structure, culture, and internal politics of the organization. The software engineer needs to be sensitive to these, since, in general, new software should not force unplanned change on the business process (Abran et al., 2005).

According to Nuseibeh and Easterbrook (2007), one of the most important goals of requirements elicitation is to find out what problem needs to be solved, and hence identify system *boundaries*. Mochal (2006) suggested that system boundaries and scope should be defined and documented because it helps all stakeholders understand the planned system. These boundaries define, at a high level, where the final delivered system will fit into the current operational environment. The identification of stakeholders – individuals or organizations who stand to gain or lose from the success or failure of a system – is also critical. One problem encountered is the fact of inhomogeneous users. Therefore the elicitation process needs to identify the needs of different user classes, such as novice users, expert users, occasional users, disabled users, and so on (Nuseibeh and Easterbrook, 2007).

2.4.3.2 Requirements elicitation techniques

There are various techniques available for requirements elicitation and the selection depends on the time and resources available to the requirements engineer, and of course, the kind of information that needs to be elicited.

Requirements elicitation requires the selection of appropriate practices, methods and techniques for gathering requirements. Selecting appropriate techniques for gathering requirements is not easy. Since there is no universal technique that can be used for all situations and each situation is unique, system developers need to identify which technique (or techniques) best suits a specific project. Many techniques are available and each one has advantages and disadvantages differently.

Nuseibeh and Easterbrook (2007) have distinguished a number of classes of elicitation techniques as following.

2.4.3.2.1 Traditional techniques

Traditional techniques include a broad class of generic data gathering techniques. These include the use of questionnaires and surveys, interviews, and analysis of existing documentation such as organizational charts, process models or standards, and user or other manuals of existing systems.

According to Coulin and Zowghi (2005) and Ding (2005), interviews are probably the most traditional and commonly used technique for requirements elicitation because interviews are essentially human-based social activities, inherently informal and their effectiveness depends greatly on the interaction between the participants. Interviews give an efficient way to collect large amount of data quickly. The effectiveness of interviews, however, depends on the skill of the interviewer and the usefulness of the information gathered (Goguen and Linde, 1993).

Typically there are two types of interviews: unstructured and structured interviews.

Unstructured interviews are conversational in nature where the interviewer enforces only limited control over the direction of discussions (Coulin and Zowghi, 2005). Unstructured interviews carry the risk that some topics might be completely neglected since they do not follow a predetermined agenda or list of questions. Maiden and Rugg (1996) have added that a typical problem with unstructured interviews is to focus in too much detail on some areas and not enough in others. It is suggested that this type of interview is best applied when there is a limited understanding of the domain or as a precursor to more focused and detailed structured interviews (Coulin and Zowghi, 2005).

Structured interviews are carried out based on a predetermined set of questions to gather particular information. Effective structured interviews have to address the right questions to be asked, when should they be asked, and who should answer them. Templates that provide guidance on structured interviews for requirements elicitation such as Volere (Robertson and Robertson, 1999) can be used to support this technique. Although structured interviews seem to limit the investigation of new ideas, they are generally considered to be rigorous and effective (Coulin and Zowghi, 2005).

However a significant previous research by Boonramphai (2008) has found that a semi-structured interview technique (a mixture between unstructured and structured interviews) was a standard technique used in many knowledge engineering projects in Thailand. Semi-structured interview technique makes use of a redesigned set of questions but allows unplanned supplementary questions to be asked during the session (De Waele and Harré, 1979).

In addition to interviews, it is likely that questionnaires are usually used to collect information from multiple stakeholders because it is quick. However, questionnaires have a limitation in the depth of knowledge they are able to elicit. The opportunity to investigate further on a topic or expand on new ideas is lacking with questionnaires. In addition, questionnaires provide no mechanism for the respondents to request clarification or correct misunderstandings. Coulin and Zowghi (2005) have suggested that questionnaires should be used during the early stages of requirements elicitation. They are useful as informal checklists to ensure fundamental elements are addressed early on and to establish the foundation for subsequent elicitation activities. However, the terms, concepts, and boundaries of the domain must be well established and understood by the participants and questionnaire designer in order to create effective questionnaires. Questions have to be focused in order to prevent gathering large amounts of redundant and irrelevant information.

Finally, the existing documentation, the existing system will provide formally defined requirements for the new system in order to help them understand the organization and prepare other kinds of fact finding. Existing documentation includes:

- organization documents (e.g. business forms, work procedures, job descriptions, policy manuals, business plans, organizational charts, correspondence, minutes, accounting records, external correspondence, customer complaints)
- system forms and reports (e.g. computer screens and reports, system operating manuals, user and technical documentation, system analysis and design models)
- business domain journals and reference books
- software packages

However, existing documentation often does not match up to reality since it is out of date, and may not reflect the differences between policy and practice.

Karten (1994) has offered several suggestions to extract information from customers:

- Take nothing at face value. Never make assumptions. Repeat questions already asked, rephrasing them to get different perspectives.
- Ask for clarification. Be sure to understand the customer's language used to describe a problem. Do not be afraid of admitting not knowing something. Stress to the customers how important the questions are to ensure the developer fully understands their expectations.
- Gather information from multiple sources. A broader perspective of diverse needs can be obtained by presenting similar questions to individuals in separate interviews. Multiple responses help to fill in the gaps in what any one given source provides.
- Watch for inaccuracy. Be sure to consider the skill-level and any resistance in the customer. Are they answering questions from their own perspective or simply trying to tell the questioners what they want to hear? (Karten, 1994).

2.4.3.2.2 Group elicitation techniques

Group elicitation techniques are another important tool for requirements elicitation. This is in agreement with Coulin and Zowghi (2005) that group work is a well-established and often-used technique in requirements elicitation. This technique is particularly effective since it would appear that groups are able to deal with complex tasks such as requirements elicitation better than individuals because they have a wider range of skills and abilities to draw from (Coulin, 2007). Group elicitation techniques aim to foster stakeholder agreement and buy-in, while exploiting team dynamics to elicit a richer understanding of needs (Nuseibeh & Easterbrook 2000). Moreover, group techniques are naturally very important to the requirements elicitation process because software development is inherently a group effort (Palmer & Fields 1992). They include brainstorming and focus groups, as well as Joint Application Development (JAD) and Rapid Application Development (RAD) workshops (using consensus-building workshops with an unbiased facilitator) and organizational game (Maiden and Rugg, 1996).

Saiedian and Dale (2000) have reported the use of organizational game, demonstrated by The Aarhus University Research Foundation during its PACS project with Skejby Hospital in Aarhus, Denmark. In conducting the organizational game, situation cards - index cards describing a particular situation at the workplace - were used. These situations were solicited from the customer prior to the activity and described situations in which normal procedures were not followed. During the game, the situational cards were employed to trigger discussions about how particular situations are actually handled and other ways they could be handled (Kjaer and Madsen, 1995, p.7). Such discourse assists the developer in identifying requirements raised by exceptions.

Brainstorming involves participants from different stakeholder groups engaging in informal discussion to rapidly generate as many ideas as possible without focusing on any one in particular (Coulin and Zowghi, 2005). The advantage of this technique is to promote freethinking and expression, and allows the discovery of new and innovative solutions to existing problems. This technique therefore is suggested to be used for developing the preliminary mission statement for the project and target system. This technique however is not suitable for resolving major issues or making key decisions. It is inappropriate for exploring or critiquing ideas in great detail.

According to Garmer et al. (2004), focus group interviews are important in order to, e.g. incorporate contextual issues and to highlight and rank problems that the end-users' experienced and regarded as the most important problems to be solved. Requirements workshops are also a powerful technique for eliciting requirements because they can be designed to encourage consensus concerning the requirements of a particular capability. They are best facilitated by an outside expert and are typically short (Young, 2002).

2.4.3.2.3 Prototyping technique

Prototyping has been used for elicitation where there is a great deal of uncertainty about the requirements, or where early feedback from stakeholders is needed (Davis, 1992). It is involved with providing stakeholders with prototypes of the system to support the investigation of possible solutions is an effective way to gather detailed information and relevant feedback (Coulin and Zowghi, 2005). Uncertain requirements can also come from customers who are uncertain of exactly what they want or need. System engineers therefore might get only a "wish list" of what the customers would like to see, with few details or without being clear as to whether the list is complete (Pfleeger and Atlee, 2005). Moreover, it is easy for many people to criticize, in detail, an existing product than to image, in detail, a new product. As such, one way that we can elicit details is to create a prototype of the proposed system and to solicit feedback from potential users about what aspects they would like to see improved (Pfleeger and Atlee, 2005).

This technique provides the benefit of encouraging stakeholders, and more specifically the users, to play an active role in developing the requirements. Prototyping is a technique which is widely used when the requirements are not clear. It is process of developing a rough replica of the proposed system so as to get a clear understanding of the requirements (Austin, 2007). It is the construction of an executable system model to "enhance understanding of the problem and identify appropriate and feasible external behaviors for possible solutions" (Hsia et al., 1993, p. 77). Using prototyping technique can help in determining whether the customer's problem has a feasible solution, or exploring options for optimizing quality requirements.

Prototyping is the development of a proposed system replica that can be easily modified and normally includes the proposed systems interface and the input/output (I/O) functionality. Before the commencing on development it is fairly better to gather requirements explicitly, organize them and translate them into a system requirements specification document after validation. This way the product developed will be of good quality and acceptable by the customer.

Prototyping provides a visual representation of requirements and are of different kinds and may include painted screens, paper representations, animated screens or even programs (Mahendran, 2006). There are two main types of prototypes: (i) throw-away prototyping, and (ii) evolutionary prototyping.

Throw-away prototyping is prototype that is developed as a non-functional model of the proposed system so that developers can gather the user requirements in unclear situation and after the requirements are consistent and structurally complete, the prototype is discarded (hence the name 'throw away') (Acosta et al., 1994). From there the system can be developed using other development methodology. The objective here is to derive and validate system requirements.

The developer starts with those known and poorly understood user requirements. After creation of the prototype it is presented to the user (in users working environment) so that they can make suggestions of improvements of structure/ features, additions or removal features. The views are then incorporated to the prototype. The iterative process continues until a final requirement agreement is reached at which level requirements are explicitly consistent and complete, then the prototype is thrown away.

The final prototype should not be used as the final system because it does not fully model all the system characteristics and typically has no specification for long term maintenance because the system is poorly structured and hard to maintain. The prototype is normally undocumented. The advantage of the exercise is that throw-away prototypes can be developed quickly and the method used is informal. The model is used by users to re-examine their needs and clarify their requirements. The model is thrown away after real requirements have been elicited.

Evolutionary prototyping is an approach to prototyping where a fully working very robust and structured system prototype is initially produced and then refined through a number of phases of software development until it becomes the final system. The objective of this kind of prototyping is to produce and deliver a fully and correctly working system to the customer. The initial prototype forms the heart of the system and changes will be built on it.

The process starts with those requirements that are best known and understood which are used to develop the first working prototype. This working system is presented to the user by a qualified or skilled prototype developer who acts in the place of the user, watches how users operate it and difficulties they encounter. Then takes these views together with those of the users, documents them as requests to be incorporated to the prototype during the next stage of development. This continues until the final system is delivered to the customer or end users. Evolutionary prototyping is used for systems where advance development of specifications is impracticable. Techniques used allow rapid system iterations. Since, there are no specifications then the verification of the system is not possible.

Advantages of evolutionary prototyping approach are:

- System development acceleration
- Intensive user involvement –the system developed is more likely to meet user needs and satisfy all stakeholders involved. The intensive user involvement will help users to commit to the use of the system.
- The phases of development are done at the same time (specification, design and implementation)
- The prototypes developed at each stage are deliverables to the customer.
- User interface is likely to be of high quality

Problems of Evolutionary prototyping include:

- Most organizations standardized management processes tend to take waterfall model approach to system development.
- The approach requires a highly skilled team of development that may be hard to get
- The approach entails continued changes which iteratively damage system structure making it hard and expensive to maintain such a system.

Users may use the prototype on an interim basis awaiting delivery of the refined system. Another approach that developers can opt to proceed with when the requirements are unclear is the incremental approach to development (Aldaijy, 2004). In this approach, the proposed system is developed and delivered to customers in increments after establishment of overall system architecture.

The requirements and specifications are developed for each increment separately and users use these increments for experimentation while at the same time other increments are being developed. This way those known requirements can be used to start off the development and other requirements gathered afterwards for each increment during increment development. This approach combines same techniques and advantages of prototyping that ensures delivery of a system that conforms to user requirements.

2.4.3.2.4 Model-driven techniques

Model-driven techniques provide a specific model of the type of information to be gathered, and use this model to drive the elicitation process (Tuunanen, 2003). These include goal-based methods, such as KAOS (van Lamsweerde et al., 1998) and I* (Chung et al., 2000), and scenario-based methods such as CREWS (Maiden, 1998). This technique requires a thorough knowledge of the domain area of the system or a high level of knowledge of work practices which the end-users hardly posses (Tuunanen, 2003).

A scenario is one of several techniques used for requirements elicitation. Scenarios are defined as "informal narrative descriptions…stories about human activity", and are used "to conduct analysis and design in a vocabulary that permits end-user participation" (Carroll, 2000). Scenarios are often particularly defined situations including a particular setting, a central character or "agent" and a plot, consisting of a sequence of actions and events (Carroll, 2000). Cox (2008) stated that the requirements discovery phase is less served by scenario approaches than other phases.

Scenarios are useful for requirements gathering because they permit people to discuss situations without reference to specific technologies. This is particularly valuable when working with non-professional people as their frequent lack of knowledge about technical language and different technologies can be often prove a barrier in requirements gathering (Eisma et al., 2003).

2.4.3.2.5 Cognitive techniques

Cognitive techniques include a series of techniques originally developed for knowledge acquisition for knowledge-based systems (Shaw and Gaines, 1996). Such techniques include protocol analysis (in which an expert thinks aloud while performing a task, to provide the observer with insights into the cognitive processes used to perform the task), laddering (using probes to elicit structure and content of stakeholder knowledge), card sorting (asking stakeholders to sort cards in groups, each of which has name of some domain entity), repertory grids (constructing an attribute matrix for entities, by asking stakeholders for attributes applicable to entities and values for cells in each entity).

Protocol analysis is where participants perform an activity or task while talking it through aloud, describing the actions being conducted and the thought process behind them (Coulin and Zowghi, 2005). According to McGraw and Harbison-Briggs (1989), this technique can provide the analyst with specific information on the actual processes the target system must support. However talking through an operation is not the normal way of performing the task in many cases; as a result, it may not necessarily represent the true process completely or correctly.

Laddering involves stakeholders being asked a series of short prompting questions, known as probes, and requiring them to arrange the resultant information into an organized hierarchical structure (Coulin and Zowghi, 2005). This technique will work effectively when stakeholders are able to express their understanding of the domain and arrange that knowledge in a logical way, which hardly happens.

For card sorting technique, stakeholders are required to sort a series of cards containing the names of domain entities into groups according to their own understanding, and to explain the rationale for the way in which the cards are sorted (Coulin and Zowghi, 2005). Card sorting will be effective when all entities are included in the process and this can occur only when the domain is sufficiently understood by both the participants and the analyst. This is often difficult to achieve. Both card sorting and laddering techniques are mainly used as a way to clarify requirements (rather than elicit requirements) and categorize domain entities (Coulin and Zowghi, 2005).

According to Kelly (1955), repertory grids concerns with asking stakeholders to develop attributes and assign values to a set of domain entities. As a result the system is modeled in the form of a matrix by categorizing the elements of the system, detailing the instances of those categories, and assigning variables with corresponding values to each one (Coulin and Zowghi, 2005). This technique represents a level of abstraction unfamiliar to most users since the objective is to identify the similarities and differences between the different domain entities. Therefore, this technique is generally employed when eliciting requirements from domain experts.

2.4.3.2.6 Contextual techniques

Contextual techniques emerged in the 1990's as an alternative to both traditional and cognitive techniques (Goguen and Linde, 1993). These include the use of ethnographic techniques such as participant observation. This involves the analyst overtly or covertly participating in the normal activities of the users over an extended period of time whilst collecting information on the operations being performed (Coulin and Zowghi, 2005). They also include ethnomethodogy and conversation analysis, both of which apply fine grained analysis to identify patterns in conversation and interaction (Viller and Sommerville, 1999). However, generally ethnography is very expensive to perform and requires significant skill and effort on the part of the analyst to interpret and understand the actions being performed.

In conclusion, no one technique in isolation is able to capture all the requirements completely (Maiden and Rugg, 1996). Also, each technique has its advantage and disadvantages, and is normally best suited for use in particular application domains. The requirements engineer needs to select the appropriate technique since an ineffective technique can cause project delay or over budget.

2.4.4 Key players and their roles in requirements elicitation

People are involved in both the main contexts of requirements elicitation: requirements sources and elicitation techniques. Saiedian and Dale (2000) have identified the key players and their perspectives/roles in requirement elicitation as discussed in the following section.

2.4.4.1 Key players from the customer side

According to Saiedian and Dale (2000), the key players (from the customer side) and their respective perspectives/roles can be classified as follows:

- *Buyer:* buyers are the people responsible for contracting and paying for the software system. Their chief concerns include project schedule and budget, even to the point of compromising usability. These individuals are usually the project point of contact for meetings with the developer. While their background may be technical in nature, they are often long removed from the user/domain expert relationship with the work environment they oversee.

- *End user*: end users are the individuals who ultimately will use the system developed. As such, they are most concerned with the usability, reliability, and availability of the system. These individuals are the ones most familiar with the specific work procedures being addressed by the system. They have the greatest stake when it comes to issues concerning user interfaces and user guide documentation.

- *Domain experts*: these are the individuals who understand the system environment or problem domain where the software system will be employed. They are the source for technical input regarding system interface detail and requirements.

- *Software maintainers*: for projects that will eventually be maintained by the customers, these are the individuals who will be responsible for future change management and implementation and anomaly resolution. As such, they are most interested in internal product issues such as design documentation and system architecture (Saiedian and Dale, 2000).

2.4.4.2 Key players from the developer side

According to Saiedian and Dale (2000), the key players (from the developer side) and their respective perspectives/roles as appropriate can be classified as follows:

- *Program management/Manager*: these are the individuals responsible for product sales and marketing as well as overall project development oversight. Often, these are the individuals who deal directly with the customer.

- *Requirements engineers*: these are the individuals, usually system engineers, who are responsible for the identification and documentation of the requirements.

- *Software engineers*: these are the individuals who provide expertise on software design constraints, prototype development, and technical feasibility.

- *Testers*: testers are responsible for developing and executing the necessary test conditions for development and sell-off activities. These include module tests, integration tests (as elements of the product are brought together), and ultimately system-level functional tests (both standalone and integrated with external interfaces). All are designed to validate and demonstrate delivered capability in level-up fashion. As such, testers must intimately understand and trace requirements from origin to final product to ensure product validation is feasible and completely and accurately documented. If we cannot adequately demonstrate to the user that the product meets their needs at sell-off, then its acceptance will be jeopardized (Saiedian and Dale, 2000).

For effective requirements information gathering, it is important that these various disciplines be organized and coordinated into a team for brainstorming and analysis activity with the customer (Saiedian and Dale, 2000). A requirement gathering requires the developer representatives to have a wide variety of knowledge and skills in addition to the specific technical expertise in the application domain and software development. These include interviewing, group work, facilitation, negotiation, problem solving, and presentation skills.

2.4.5 Problems and difficulties with requirements elicitation

The elicitation of requirements is not an easy process, and therefore it confronts many problems. The most common problems that hinder the identification/definition of the user's needs are proposed by Saiedan and Dale (2000) and summarized in the following.

2.4.5.1 Unclear requirements

Situations where requirements are unclear or are poorly understood arise due to complexity of problem at hand and also human complexity. Problems that entail complex and large real world entities require solutions of complex and large systems. It is hard or difficult to know precisely the requirements of such systems.

Many times, especially in an organization whose users don't know much about computers, requirements provided by the users are often unclear, ambiguous, contradicting and also incomplete. Proceeding with system development when requirements are unclear would mean that developers would guess what the system is required to achieve and then start development. This will result in a situation where they will return later to the requirements engineering cycle and rework the software when they discover discrepancies. Therefore, developers need to develop and adapt a system development lifecycle methodology that will help to better capture and understand user requirements.

A commonly used methodology in development when needs are not clear is the prototyping approach. Prototyping approach is combined with requirement management tools whose main focus is on requirement change management and requirement traceability.

2.4.5.2 Poor communication

User requirements are the basic statements of what the users want to do with the system from the user's point of view. To gather these requirements or needs, developers need to interact and communicate with users of the proposed system. The interactions and communications are aimed at requirements gathering by use of questionnaires, interviews, observations, storyboarding, prototyping, workshops and even analysis of available documentation of current system or systems with the same domain.

According to Karten (1994, p.38), although we can communicate goals, objectives, tasks, procedures, constraints, interdependencies, timetables, priorities, responsibilities, accountabilities, and even deliver the solution that perfectly meets our customer's needs—and still not meet our customers expectations. That is because how we communicate can be just as important as what we communicate.

If requirements engineers unintentionally exhibit signs of not-listening when they work with customers, the customers' impression that the requirements engineers are distracted or uninterested is created. As a result, the ability to work together is undermined. In contrast, if customers' perception is that the requirement engineers are listening, it leads them to feel comfortable with the engineers and to open up to the engineers.

In both verbal and written communication, it is important not to contradict what is intended to be communicated. Over-eagerness to be responsive or the need to match or exceed the level of performance by a competitor can cause requirement engineers to promise something that they cannot deliver (Saiedian and Dale, 2000). Conflict can also occur if a level of service excess what is promised. For example, if the project is finished before the set schedule or delivered more than promised, the customers will begin to expect the same in future projects. Moreover, they may suspect that the developers are padding the cost and schedule estimates in their proposals.

Indirect communication links are those in which the customer and developer do not deal with one another directly but communicate through intermediaries. The marketing and sales link (in which a salesperson serves as the intermediary) is one example of this as is a MIS intermediary who defines corporate customer's goals and needs to designers and developers (Saiedian and Dale, 2000). Often intermediaries do not completely understand the customer's needs. As a result, they can intentionally or unintentionally filter and distort messages.

In addition, communication preferences often vary from customer to customer and from one circumstance to another. Each has its own priorities, time frames, pressures, and objectives (Saiedian and Dale, 2000). It is the responsibility of the developer to be cognizant of and adapt to the customer's communication styles considering such things as their pace of activity, receptiveness to new ideas, level of risk-taking, adherence to protocol, and written and oral presentation formats (Karten, 1994). Showing an interest in the customer's preferences can imply them that they are important. This ultimately facilitates the relationship supporting a better communication necessary for working together. It is important to consider customer preference before they make a complaint.

2.4.5.3 Resistance

Resistance is a physical expression of an emotional process-taking place within a person and takes the form of opposition (Saiedian and Dale, 2000). Resistance to new ideas normally occurs for any improvement process. Nobody likes being told they are obsolete or that they can do things better. Moreover, technological "revolutions." are often too frequent for most people's taste, especially when the costs of learning a new technology come close to outweighing the benefits (Borenstein, 1991, p. 42).

Being able to recognize and mitigate resistance is critical for the requirements engineer because it highlights issues important to the customer that are not being addressed adequately. Some of the more common forms of resistance are listed and briefly described below (Saiedian and Dale, 2000):

- *Time resistance.* Person never has time to meet.
- *Overload resistance.* No matter how much information is given to the person, it is never enough.
- *Silence resistance.* Person does not react or respond to anything said.
- *Impracticality resistance.* Person always reminds you that he lives in the real world.
- *Compliance resistance.* Person always agrees with you. Reservations are never expressed and the implications that whatever you do is fine (Saiedian and Dale, 2000).

2.4.5.4 Articulation/expertise problems

According to Saiedian and Dale (2000), a common obstacle to communication on technical matters is the use of terminology that is not understood by one of the parties, particularly by customers. Professionals like to use jargon and acronyms profusely and this can cause confusion, and can annoy, or intimidate the customers. It is therefore important to adjust this to the customer's level of technical sophistication.

The complication of the elicitation process is also exacerbated by the complexity of modern software systems. Many systems have vast interconnections between subsystems and even environments that even professionals in specialized disciplines hardly understand. Even more problematic is the ever-changing nature of requirements as user's learn and grow, together with the difficulties of integrating multiple, diverse and conflicting views (Saiedian and Dale, 2000).

Another problem can be the developer "talking down" to the "dumb users" (Saiedian and Dale, 2000). In the minds of some engineers and programmers, making software easy to use is necessary only because "most people are too unsophisticated to understand the glories of a real computer system" (Borenstein, 1991). When a system is approached with that attitude, the result can be downright offensive to the oft-intelligent user (Saiedian and Dale, 2000). It is quite often that many people think that something is too complex to explain, and so consider explaining it a waste of effort. However, taking time to educate the customer has a number of advantages. It not only forces developers to clearly understand the problems and how their solutions address it, but it explains that explicitly to the user. More importantly, it shows a confidence in their works and willingness to go above and beyond to help the customers. This in turn can help build trust with the customers and encourage them to be more disclosure.

An organizational problem is that project managers fail to acquire or deploy experienced requirements analysts (RAs), and that RAs are not sufficiently trained and/or experienced to perform their roles effectively (Whitten, 2005). Young (2004) suggested the fields of training that should be considered:

1. The importance of requirements to project success, based on industry experience
2. The value of good requirements
3. Roles, skills, and characteristics of an effective RA
4. Having and using a requirements process

5. The value of investing more in the requirements process (8% to 14% of total project cost)
6. The project's requirements process
7. Overview of the mechanisms, methods, techniques, and tools that will be used;
    - Types of requirements
    - The requirements repository (and its many components);
    - Gathering requirements – the techniques to be used
    - Writing good requirements
8. Reducing rework on the project
    - Identifying the real requirements
    - Controlling changes to requirements and the addition of new requirements

Inappropriate technique selection is another problem. According to Hickey, and Davis (2003), it was found that less experienced analysts often select a technique based on one of two reasons: (i) it is the only one they know, or (ii) they think that a technique that worked well last time must surely be appropriate this time. As a result of such inappropriate selection, less experienced analysts seem to be less successful in uncovering the user needs than extensive experienced analysts.

2.4.5.5 Problem perspective differences

Most firms in the high-tech industry have not yet used customer-centered techniques for their front-end development processes. Moreover, most developers hardly have experience as an end-user in the application domain for which they develop software. As a result, development tends to be technology (or solution) driven without a contextualized sense of the problem to be solved. Many times products are offered to customers based on the whims of what updated hardware and software packages allow developers to do, rather than based on the actual needs of customers. Borenstein (1991) has offered a great example of this misguided technology fixation that:

"The developer may emerge wild-eyed from his office, ranting to anyone who will listen about the breakthrough he has made, about how easy his new gadget or program is to use, about how it will revolutionize the way people talk to computers. His colleagues will smile uneasily and shift restlessly from one foot to another, pondering several alternatives, all of them unpleasant:

1.1 The developer has finally gone over the edge, lost his mind, and will not longer be useful for anything.

1.2 The developer has re-invented the wheel, and will eventually have to face the disappointment of knowing that his technique isn't new at all.

1.3 The developer has indeed made a minor innovation, but nobody is really likely to care about it because it is irrelevant for most purposes.

1.4 The developer is telling the truth, but the world is not ready for his breakthrough, and it will languish without aggressive and expensive marketing.

1.5 The developer is telling the truth, and the world is about to sit up and take notice. The colleague is going to have to take the time it takes to learn how to use this new breakthrough. Since he is not (yet) aware of any need for the breakthrough, this is an unappealing prospect"

In contrast, if customers and end users are knowledgeable in the application domain but not in the process of engineering software, they might not be aware of any existing limitations in their process that recent technology improvements have alleviated. They simply do not know what it is they can ask for in a new or updated product development. It is therefore the developers' responsibility to inform them of the possibilities. However for most developers business strategy requires making money. It is therefore much more appealing to present a solution that requires significant new development effort than a simple modification to an existing tool. When developers create a system that is not truly needed, the strategy can backfire and potentially lose them future contracts. Therefore in order to inform customers or end users of the possibilities, developers need to keep the focus on the benefit to their activities as well.

This chapter has briefly reviewed the literature pertaining to requirements and requirements engineering. It will form the basis of comparison of data sourced via surveys in this research. References in such comparison to "theory" or "best practice" will mostly be to the information discussed in this chapter.

CHAPTER 3
RESEARCH METHODOLOGY

This chapter will describe the research methodology used in this study. Research procedure, research instrument, population and sampling, collection of data, and data analysis are discussed in this chapter.

3.1 Research Procedure

The research procedure for this study is shown in Figure 3.1. First the purpose of the research study was agreed to. Then designing the study in terms of questions to ask, and research methodology were determined. The next step was the collection of data. Data used in this research consisted of both primary data and secondary data. The primary data was collected, using questionnaires, from system engineers/software engineers/requirements engineers who work for software and construction companies in Thailand. The secondary data was collected from the related literature review of several sources such as textbooks, academic journals, and websites. Then the primary data was processed using the software tool Statistical Program for Social Science (SPSS) while the secondary data was classified and combined with the primary one for further analysis. After the analysis of both primary and secondary data, conclusion of the research and recommendation was finally formulated.

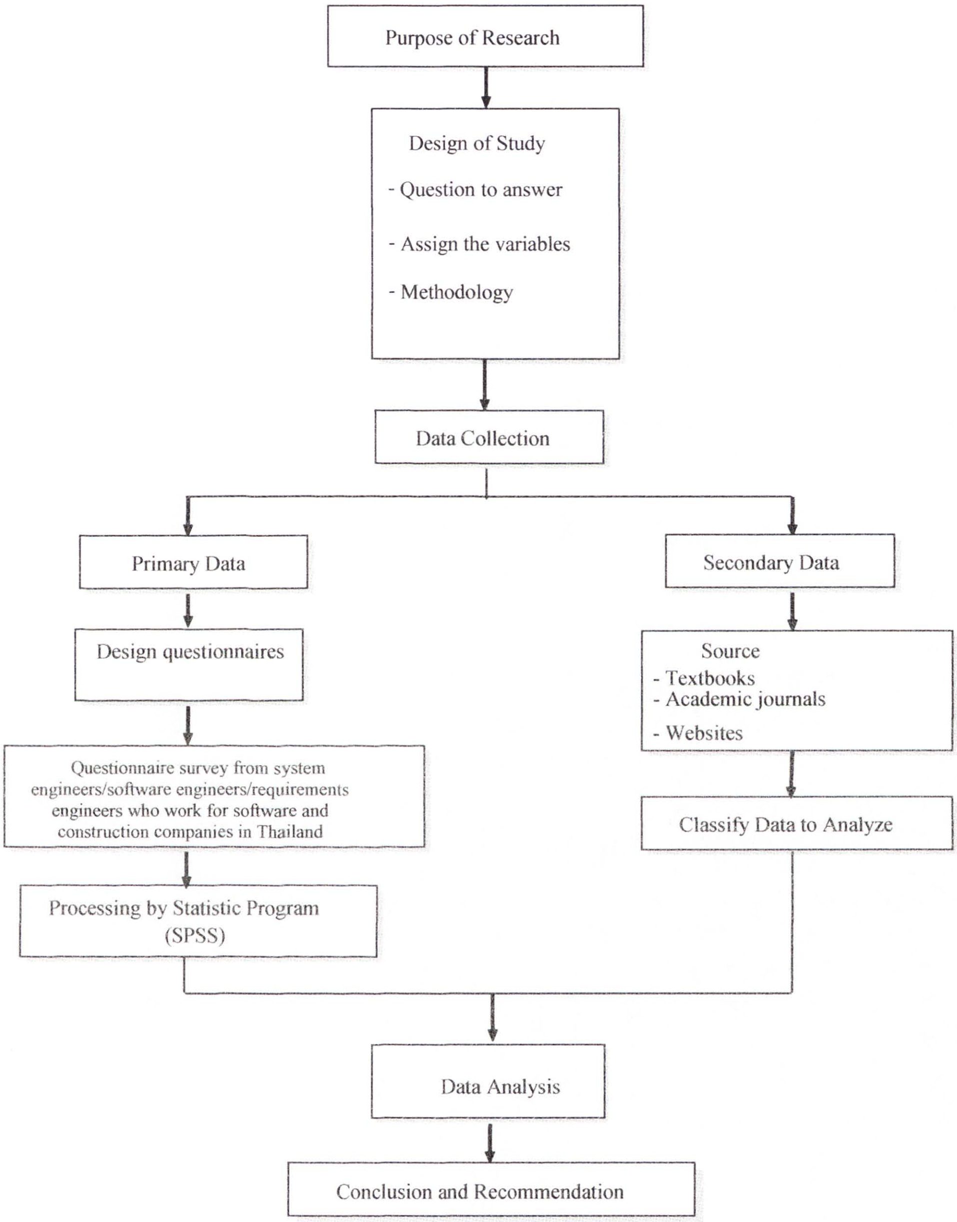

Figure 3.1: Research procedure in this study

3.2 Research Instrument

Secondary and primary researches were considered for use in collecting data. The former was the literature review from various sources, and the latter was via questionnaire survey (quantitative method).

3.3 Secondary Research

Veal (2003); Saunders *et al.* (2003) have define secondary data as data that is already existent and collected for some other objectives. It is work that has already been carried out on a particular subject (Finn *et al.*, 2000). In this case, the researcher became the secondary user of the data.

In this research, secondary data were utilised for two reasons. Firstly, to provide the researcher with topic ideas for this research by exploring the previous researches regarding the concept of requirements engineering, particularly, requirements elicitation. The second reason was to expedite the research process. There were many kinds of secondary data available to the researcher. The secondary data needed in this research was gained from academic journals, websites, textbooks, related research projects and also the internet. According to Zikmund (1997), researchers should start by examining secondary data before collecting their own. The time taken to obtain significant secondary data with regard to requirements engineering and requirements elicitation was about three months.

The advantages of secondary data include convenience of access to the data and the relatively low cost for obtaining a variety of information. Furthermore, collecting information from the literature review leads to the creation of innovative ideas from the previous research (Oppenheim, 1992). In addition, secondary data analysis can be used to merge data from various sources in order to provide wider and more useable data sets to confirm that data are compatible (Finn *et al.*, 2000). Moreover, the experience of those who collected the original data can be manipulated (Veal, 2003). However, the disadvantage is that the availability of documents may be limited with the specific aims of the study (Veal, 2003). Consequently, the data requires careful consideration of ways in which it might be used.

3.4 Primary Research

Primary data is the data that researchers have to collect on their own by doing primary research (Veal, 2003). Primary research basically comprises quantitative and qualitative methods. Primary research in this research focused on only quantitative (questionnaires) method.

3.5 Quantitative research

According to White (2000), surveys using questionnaires where responses are given numerical values are referred to as quantitative research. Quantitative data is, therefore, non-figurative data, they are hard and reliable; they can be measured from tangible and countable data (Bouma and Atkinson, 1995 in Naoum, 1998). The quantitative method consists of survey questionnaires which are of two types: self-completion (respondent-completion) and interviewer completion questionnaires (Jennings, 2001).

The advantages of quantitative methods are they allow simple mathematic analysis, and the results can be represented as pie charts, tables and bar charts and there is confidence in mathematic and statistical procedures (White, 2000).

The questionnaire method was used in this study to conduct the primary research. This research utilised respondent-completion since it was cheaper, quicker and anonymous.

3.5.1 Questionnaire

A questionnaire survey refers to a series of questions, each one providing a number of alternative answers from which the respondents can select (White, 2000). This research is conducted by such a questionnaire. It presents the fact or information and opinions from a large number of customers (Jennings, 2001). Questionnaire survey is treated as a quantitative method allowing for the analysis of quantities or measures within the collected data (Veal, 2003). Use of self-completion (respondent-completion) type survey avoids interviewer bias (White, 2000). Both closed-end and opened-end questions are used in the questionnaire.

3.5.2 Sampling Procedures

Samples for this research were selected based on criteria explained in Chapter 1. The researcher chose 60 companies who do construction and software businesses (30 each). These software firms have done software projects including IT. The selected samples are medium- and large-sized firms. The medium-sized firms are those which have investment capital between 50-200 million Baht and 50-200 employees while large-sized ones have more than 200 million Baht and more than 200 employees.

The researcher sent an introductory letter and questionnaires to the companies to ask for their participation in this research. Self-administered questionnaires were used. Return envelopes were enclosed for returning questionnaires back to the researcher.

3.5.3 Pilot test

Pilot testing is an important tool to test and improve the reliability and validity of the questionnaires. Saunders et al. (2003) have stated that the aim of the pilot test is to refine the questionnaire so that respondents will have no problems in answering the questions. It also improves content validity since suggestions can be made on the structure of the questionnaire (Saunders et al., 2003). Pilot test lets us know how long it takes to complete the questionnaire and which questions are unclear, ambiguous and uneasy for respondents.

The number of questionnaires used for the pilot test in this research is 30 and the respondents are software engineers/system engineers/requirements engineers who work for software and construction companies

3.6 Data analysis of quantitative data: Statistical Software Packages for Social Sciences Version 12 (SPSS)

In this study, the quantitative data analysis was used employing the tool Statistical Software Packages for the Social Sciences (SPSS) version 12 and descriptive statistics were used to identify the nature and characteristic of all the customers and analyse the research results.

CHAPTER 4
SUMMARY OF SURVEY RESPONSES

This chapter will present the results obtained from the survey on 60 companies who do construction and software business in Thailand (30 each). These firms are medium and large-sized firms. The medium-sized firms have investment capital between 50-200 million Baht and 50-200 employees while large-sized ones have more than 200 million Baht and more than 200 employees. These firms were selected because they are directly involved with requirement engineering, particularly the requirements elicitation process. The size of these companies was also large enough to be involved with projects requiring the use of requirement elicitation processes, and they used formalized process of project management and requirement engineering.

This chapter consists of five main sections:

(i) Business information of the firms who were surveyed in this research

(ii) Demographic information about the respondents who worked for these firms, and who were involved with the process of requirements elicitation

(iii) Requirements elicitation in general

(iv) State of requirement elicitation in Thailand

(v) Problems of requirement elicitation in Thailand.

4.1 Business Information

The firms that participated in this study are involved either in software development or construction business. The business context (national or international) of the 30 software development firms (50.0%) and 30 construction firms (50.0%) is presented in Table 4.1.

Table 4.1: Level of doing business of the firms who joined this research

Level of doing business	Numbers of firms that do		Total
	software development	construction	
National organization (operate only in Thailand)	17 (56.7%)	21 (70.0%)	38 (63.3%)
International organization	13 (43.3%)	9 (30.0%)	22 (36.7%)
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

The size of the firms participating in this study is presented in Table 4.2.

Table 4.2: Size of the firms who joined this research

Size of the firms	Numbers of firms that do		Total
	software development	construction	
50-200 employees	16 (53.3%)	12 (40.0%)	28 (46.7%)
More than 200 employees	14 (46.7%)	18 (60.0%)	32 (53.3%)
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

The investment capital of the firms participating in this study is presented in Table 4.3.

Table 4.3: Investment capital of the firms who joined this research

Investment capital	Numbers of firms that do		Total
	software development	construction	
50-200 million Baht	15 (50.0%)	13 (43.3%)	28 (46.7%)
More than 200 million Baht	15 (50.0%)	17 (56.7%)	32 (53.3%)
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

The age of the firms participating in this study is presented in Table 4.4. The result implied that in average construction firms in Thailand have been established for a longer time than software development firms.

Table 4.4: Time of establishment of the firms who joined this research

Time of establishment	Numbers of firms that do		Total
	software development	construction	
Between 1 to 3 years	6 (20.0%)	3 (10.0%)	9 (15.0%)
Between 3 to 6 years	17 (56.7%)	17 (56.7%)	34 (56.7%)
More than 6 years	7 (23.3%)	10 (33.3%)	17 (28.3%)
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

4.2 Demographic Information

This section provides demographic information of the representatives of the firms who responded to the questionnaire presented in Table 4.5. (The "other" occupations were all engineers of some discipline)

Table 4.5: Occupation of the respondents

Occupation	Numbers of firms that do		Total
	software development	construction	
Software engineer	8 (26.7%)	-	8 (13.3%)
System engineer	11 (36.7%)	-	11 (18.3%)
Manager	9 (30.0%)	9 (30.0%)	18 (30.0%)
Supervisor	2 (6.7%)	13 (43.3%)	15 (25.0%)
Other	-	8 (26.7%)	8 (13.3%)
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

The length of involvement of the responder in requirement engineering at the various firms in this study is presented in Table 4.6. Based on this data, the duration of involvement in requirements engineering in Thailand was predominantly in the range of 3-6 years.

Table 4.6: Years of experience as software engineers and/or of involvement in requirement engineering

Years of experience	Numbers of firms that do		Total
	software development	construction	
Less than 1 year	1 (3.3%)	2 (6.7%)	3 (5.0%)
Between 1 to 3 years	7 (23.3%)	6 (20.0%)	13 (21.7%)
Between 3 to 6 years	19 (63.3%)	15 (50.0%)	34 (56.7%)
More than 6 years	3 (10.0%)	7 (23.3%)	10 (16.7%)
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

Main role and responsibility of the respondents in this research are summarized in Table 4.7.

Table 4.7: Summary of main role and responsibility of the respondents

Occupation	Main role and responsibility
Manager	- Prepare project plans, task lists, timelines, resource assignments, and related budgets. - Identify, deliver and support new technologies, systems and solutions, as well as being responsible for the delivery of agreed projects as identified by the Senior Management. - Manage a team of developers and provide hands-on development skills in design and implementation of technologies - Manage the relationships with all the stakeholders involved with the firm - Manage the delivery of high quality production software to the business and customers - Manage a team of software developers, architects, project managers, testers and analysts across different locations

Occupation	Main role and responsibility
System engineer	- Interact with the customer and take up the responsibility for the technical implementation of projects and follow-up of the daily activities - Advise and guide the customer into new technologies as well as day-to-day management of the systems - Translate user needs and higher level requirements for functions into specific software implementable and verifiable requirements and algorithms for new functions, change request activities and problem correction - Establish an accurate understanding of the customer's concerns and requirements - Define and use of engineering tools for analysis and verification, environmental modeling, test requirement development and validation
Software engineers	- Interact with the customer and take up the responsibility for the technical implementation of projects and follow-up of the daily activities - Analyze users' needs and design, construct, and maintain general computer applications software and/or specialized utility programs. - Coordinate the construction and maintenance of a company's computer systems and plan their future growth. - Design, write, test and maintain features and functionality for object-oriented data management system - Perform programming and coding networks - Determine efficient coding processes and procedures and diagnose bugs and software problems - Plan, develop, witness and report on software (and some hardware) tests & verification.

Occupation	Main role and responsibility
Supervisor	- Supervise Information Services Business Systems programming staff - Make software recommendations to management - Evaluate customer needs and assign Business Systems staff duties to satisfy those needs - Develop a programme of work for the project and communicate with a wide range of people. - Supervising the building work - Manage cost and quality requirements in relation to the construction activities

4.3 Requirement Elicitation

Table 4.8 summarizes the responses when the respondents were asked whether they had ever worked in the field of requirement engineering or had been involved in requirement elicitation, in order to determine and compare working experience in the field of requirements engineering and/or requirements elicitation.

When comparing the working experience between software development firms and construction firms in Thailand, the result from Table 4.8 showed that more employees in the software development firms worked in the field of requirements engineering or were involved with requirements elicitation than those in the construction firms. This might relate to the nature of the work that software development firms often do. Requirements engineering is closely aligned with the service and industry sectors, and now with the manufacturing sector as it increasingly relies on automation (Asanghanwa et al., 2005). Software development firms can be classified as a service industry since most of the firms create and develop software for particular users. Thanasankit and Corbitt (2000) have also reported that role of requirements for software and system development could be traced to earlier studies of software engineering, rather than civil engineering.

Table 4.8: Involvement in requirements elicitation

	Numbers of firms that do		Total
	software development	construction	
Yes, always	10 (33.3%)	3 (10.0%)	13 (21.7%)
Yes, sometimes	18 (60.0%)	25 (83.3%)	43 (71.7%)
Not sure (cannot remember)	2 (6.7%)	2 (6.7%)	4 (6.7%)
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

When the respondents were asked to rate the performance of the requirements elicitation process at their organizations, the results obtained are shown in Table 4.9.

The results showed that just more than a half (53.3%) of the software development firms had good and very good performance while 16.7% had bad to very bad performance. These figures implied that the performance of requirements elicitation process of software development firms in Thailand was quite good.

For construction firms, 50.0% of them reported bad and very bad performance while only 3.3% reported good performance. These figures suggested that requirements elicitation of construction firms in Thailand was in a serious situation.

The better performance of the requirements elicitation process in software development firms compared to the construction firms might relate to the working experience of the respondents in the field of requirements elicitation. As previously stated, employees in the software development firms worked in the field of requirements engineering to a greater extent than those in the construction firms. The more the experience, the likelier the better performance

Table 4.9: Performance of the requirements elicitation

Performance rating	Numbers of firms that do		Total
	software development	construction	
Very well performed	4 (13.3%)	-	4 (6.7%)
Well performed	12 (40.0%)	1 (3.3%)	13 (21.7%)
Neutral/Unsure	9 (30.0%)	14 (46.7%)	23 (38.3%)
poorly performed	3 (10.0%)	12 (40.0%)	15 (25.0%)
very poor	2 (6.7%)	3 (10.0%)	5 (8.3%)
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

When the respondents were asked to rank sources of requirements knowledge that they acquired for system development (from "1" which is the most important source to "4" which is the least important), the results obtained are shown in Table 4.10 and 4.11 for software development and construction firms respectively. It was found that customers' or users' needs are the most important sources (with the highest numbers, respondents rated customers or users' needs as the 1st rank) of requirement knowledge for both software development and construction firms.

When comparing to software development firms, there was a slight difference in the least important source of requirements knowledge. Software development firms stated that existing system information was the least important while construction firms reported other stakeholders' needs as least important. This implied that software development firms in Thailand placed more importance on other stakeholders' needs than construction firms. This might relate to the fact that development of software is a complicated process and requires several sources of knowledge from all involved stakeholders.

Table 4.10: Ranking of sources of requirements knowledge that respondents from software development firms acquired for system development

Sources	Numbers of software development firms that gave			
	1st rank	2nd rank	3rd rank	4th rank
Customers or users' needs	**21**	6	3	-
Domain information	3	**14**	9	4
Existing system information	3	4	8	**15**
Other stakeholders' needs	3	6	**10**	11

Table 4.11: Ranking of sources of requirements knowledge that respondents from construction firms acquired for system development

Sources	Numbers of construction firms that gave			
	1st rank	2nd rank	3rd rank	4th rank
Customers or users' needs	**21**	7	8	3
Domain information	5	**28**	8	13
Existing system information	3	18	**18**	18
Other stakeholders' needs	1	7	11	**26**

Table 4.12 shows the usage of requirement elicitation technique by the respondents. This shows that traditional technique was the most frequently used technique to elicit requirements for software development firms. This was also the care with construction development firms. Therefore, it can be concluded that regardless the type of business, traditional techniques including interview, questionnaire survey, and analysis of existing documentation are the most popular techniques for eliciting requirements in Thailand.

Group elicitation technique came as the second rank. Both traditional and group elicitation techniques together accounted for 66.7% and 90% of total techniques used for software development and construction firms respectively.

Table 4.12: Most frequently used technique for requirements elicitation

Technique	Numbers of firms that do		Total
	software development	construction	
Prototyping	3 (10.0%)	1 (3.3%)	4 (6.7%)
Traditional techniques	11 (36.7%)	15 (50.0%)	26 (43.3%)
Group elicitation techniques	9 (30.0%)	12 (40.0%)	21 (35.0%)
Model-driven techniques	5 (16.7%)	1 (3.3%)	6 (10.0%)
Cognitive techniques	-	- -	- -
Contextual techniques	2 (6.7%)	1 (3.3%)	3 (5.0%)
Others	-	-	-
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

When the respondents were asked whether they ever had any problem or difficulty in the process of requirements elicitation, the results obtained are shown in Table 4.13. In general, it may be concluded that firms of each type encountered problems in requirements elicitation in Thailand.

When comparing the problem or difficulty in the process of requirement elicitation between software development firms and construction firms in Thailand, the result from Table 4.13 show that construction firms seemed to have more problem or difficulty than software development firms. This might be due to that software development firms had more systematic processes for requirements elicitation. Software development firms were more likely to be involved with projects related to system and software development than construction firms, whose main work involved with supervising the building work as shown in Tables 4.5 and 4.7. Most of the respondents from the construction firms were supervisors whose main role and responsibility was to supervise the building work.

Table 4.13: Have the respondents ever had any problem or difficulty in the process of requirements elicitation?

	Numbers of firms that do		Total
	software development	construction	
Yes	22 (73.3%)	29 (96.7%)	51 (85.0%)
No	8 (26.7%)	1 (3.3%)	9 (15.0%)
Total	30 (100.0%)	30 (100.0%)	60 (100.0%)

4.4 State of requirements elicitation

This section describes the respondents' attitudes towards states of requirements elicitation. An arbitrary threshold level was used for the descriptive analysis, asking the respondents to rate their levels of agreement on the statements related to the end users/customers requirements, stakeholder requirements, domain-imposed (functional) requirements, management support/commitment, communication, and requirements documentation. Table 4.14 offers arbitrary level for these aspects of requirement elicitations.

Table 4.14: Arbitrary level for attitudes towards states of requirements elicitation

Arbitrary level	Descriptive rating
4.20 - 5.00	Strongly agree
3.40 - 4.19	Agree
2.60 - 3.39	Neutral/Not sure
1.80 - 2.59	Disagree
1.00 - 1.79	Strongly disagree

4.4.1 End users/customers requirements

Table 4.15 represents the respondents' attitudes toward the end users/customers requirements attributes. The end users/customers requirements attributes were measured by five statements, as shown in the table.

The results showed that in the process of requirement elicitations from end users/customers of construction firms, end users/customers requirements were taken into consideration as the most important for both software development firms and construction firms with a mean of 4.25 and 3.98 correspondingly. Software/system engineers, engineers, managers and supervisors were concerned with what end users/customers want as well as what they did not want. They also regularly held discussions with the end users/customers about their requirements. Therefore, regardless of the type of business, users/customers requirements were considered to be the most important.

In addition, the results from Table 4.15 indicated that the end users/customers requirements always changed before the system development was finished. This corresponded with previous research of Thanasankit and Corbitt (2000) who found that Thai system analysts had to change the requirements often according to the users/customers requirements. However, this was not the care in construction firms. The end users/customers requirements for construction firms were relatively unchanged in comparison. This might be due to the fact that end users needs in construction firms were clearly defined than in software development firms.

Table 4.15: Descriptive analysis of end users/customers requirements attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
You regularly discuss with the end users/customers about their requirements.	4.07	.691	Agree	4.20	.761	Strongly agree
You completely obtain all the necessary requirements information from the end users/customers.	3.50	.820	Agree	3.73	.785	Agree
The end users/customers requirements are always changed before the system development is finished	4.43	.679	Strongly agree	3.03	.718	Neutral/Not sure
When you come to the process of requirements elicitation from end users/customers, you are concerned with what they do not want as well as what they want.	4.63	.490	Strongly agree	4.47	.571	Strongly agree
It can be said that in the process of requirements elicitation, end users/customers requirements are the most important	4.60	.563	Strongly agree	4.47	.776	Strongly agree
End users/customers requirements attributes	**4.25**	**.296**	**Strongly agree**	**3.98**	**.358**	**Agree**

4.4.2 Other stakeholders' requirements

Table 4.16 represents the respondents' attitudes toward other stakeholders requirements attributes.

These results indicate that in the process of requirement elicitations from stakeholders of software development firms, software/system engineers, managers and supervisors seemed to be unaware of other stakeholders' requirements, unlike end users/customers requirement. They might not have a chance to discuss with other stakeholders about their requirements, apart from end-users/customers requirements. If they had a chance to discuss with other stakeholders about their requirements, such discussion might not occur very regularly. This may be because they do not take other stakeholders requirements into consideration for the system development seriously, as can be seen from Table 4.10. Furthermore, the table indicates that construction firms seem to treat other stakeholder even more indifferently than do software development firms.

Table 4.16: Descriptive analysis of other stakeholders requirements attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
Apart from end-users/customers, you have a chance to discuss with other stakeholders about their requirements.	3.20	.610	Neutral/Not sure	2.57	.568	Disagree
You regularly discuss with other stakeholders about their requirements.	2.87	.434	Neutral/Not sure	3.13	.730	Neutral/Not sure
You take other stakeholders requirements into consideration for the system development as well as end users/customers' requirements.	2.90	.403	Neutral/Not sure	3.07	.521	Neutral/Not sure
Other stakeholders requirements attributes	**2.92**	**.344**	**Neutral/Not sure**	**2.92**	**.358**	**Neutral/Not sure**

4.4.3 Domain-imposed (functional) requirements

Table 4.17 represents the respondents' attitudes toward domain-imposed (functional) requirements attributes. The domain-imposed (functional) attributes were measured by three statements. The results showed that both software development firms and construction firms were comprehensively obtained all domain-imposed (functional) requirements for any system you have developed. Moreover, for software development firms, techniques, technology and software used for eliciting domain-imposed (functional) requirements are updated and the complexity of software system complicates the requirements elicitation process while construction firms might fail to consider these attributes.

Table 4.17: Descriptive analysis of domain-imposed (functional) requirements attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
You completely obtained all domain-imposed (functional) requirements for any system you have developed	4.03	.615	Agree	4.03	.669	Agree
Techniques, technology and software used for eliciting domain-imposed (functional) requirements are updated	3.40	.621	Agree	2.90	.548	Neutral/Not sure
The complexity of software system complicates the requirements elicitation process	4.03	.669	Agree	3.37	.556	Neutral/Not sure
Domain-imposed (functional) requirements attributes	**3.82**	**.379**	**Agree**	**3.43**	**.341**	**Agree**

4.4.4 Management support/commitment

Table 4.18 represents the respondents' attitudes toward management support/commitment attributes. The management support/commitment attributes were measured by six statements.

The results imply that management of software development firms in Thailand provided strong leadership for requirement elicitations. However, the results also indicated that training and resources for requirements elicitation implementation of both types of firms might be lacking due to weak commitment of management to training. When compared to software development firms, management support/commitment of construction firms for the requirement elicitations was much weaker.

Table 4.18: Descriptive analysis of management support/commitment attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
Management provide strong leadership and support for requirement elicitations	3.80	.664	Agree	3.10	.662	Neutral/Not sure
Management establishes requirements elicitation practices as an integral part of the system development process	4.10	.607	Agree	3.07	.521	Neutral/Not sure
Requirement elicitation processes and practices are standardized across the projects done by the company	3.70	.596	Agree	3.17	.461	Neutral/Not sure
Management at all levels of the organization supports the requirement elicitation	3.67	.711	Agree	2.97	.615	Neutral/Not sure
Management is willing to participate in meetings with end-users/stakeholders	3.57	.774	Agree	2.83	.592	Neutral/Not sure
Management is committed to provide training and resources for requirements elicitation implementation	3.37	.669	Neutral/Not sure	2.60	.563	Neutral/Not sure
Management support/commitment attributes	**3.70**	**.520**	**Agree**	**2.96**	**.379**	**Neutral/Not sure**

4.4.5 Communication

Table 4.19 indicates that in both verbal and written communications were used regularly in the firms surveyed. Both verbal and well-written communications are important for information exchange between users and analysts.

The results also show, for construction firms, goals, objectives, tasks, procedures, constraints and timetables might not be regularly communicated between the firms and end-users/stakeholders. This was probably due to the fact that there were intermediaries between the firms and the end users/customers.

Table 4.19: Descriptive analysis of communication attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
You always communicate goals, objectives, tasks, procedures, constraints, timetables to end-users/stakeholders regularly	3.47	.571	Agree	3.07	.583	Neutral/Not sure
You are always listening to what customers need in order to impress them	3.90	.548	Agree	3.77	.626	Agree
You always use verbal communication with end-users/stakeholders	4.07	.521	Agree	3.87	.629	Agree
You always use written communication with end-users/stakeholders	3.97	.669	Agree	3.53	.776	Agree
In the process of requirements elicitation, there are intermediates between you and the end users/customers.	2.47	.629	Disagree	2.70	.596	Neutral/Not sure
Communication attributes	**3.57**	**.291**	**Agree**	**3.39**	**.389**	**Neutral/Not sure**

4.4.6 Requirements documentation

Table 4.20 represents the respondents' attitudes toward requirements documentation attributes. The requirements documentation attributes were measured by six statements.

The results showed that awareness of the importance of requirement documentation seemed to be a problem in both surveyed firms in Thailand. Moreover, the results suggest that software development firms have poorer requirements documentation practice compared to construction firms.

Table 4.20: Descriptive analysis of requirements documentation attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
Requirement documents are laid out in a highly readable manner	3.37	.669	Neutral/Not sure	3.57	.504	Agree
All of the necessary information is provided in requirements documents, even if only by reference	2.93	.691	Neutral/Not sure	3.07	.691	Neutral/Not sure
Requirements documents clearly explain why the system is required	2.87	.937	Neutral/Not sure	2.43	.774	Disagree
Glossary is always included in the requirements documents	2.63	.809	Neutral/Not sure	2.17	.699	Disagree
All of the services provided and functions performed by the system are discussed in the requirements documents	2.60	.770	Neutral/Not sure	2.07	.640	Disagree
All constraints in the design are discussed in the requirements documents	2.90	.885	Neutral/Not sure	2.33	.711	Disagree
Requirements documentation attributes	**2.88**	**.607**	**Neutral/Not sure**	**2.61**	**.412**	**Neutral/Not sure**

4.5 Problems of requirements elicitation

This section describes the respondents' attitudes towards problems of requirements elicitation. The arbitrary level was used for the descriptive analysis, asking the respondents to rate their levels of agreement on the statements related to the frequency of problems. Table 4.21 indicates arbitrary level need for the frequency of problems. In this section, the respondents were also asked to rate their levels of agreement on the statements related to the problems of scope, problems at understanding, problems of volatility and problems in requirements elicitation techniques.

Table 4.21: Arbitrary level for attitudes towards frequency of problems

Arbitrary level	Descriptive rating
4.20 - 5.00	Very often
3.40 - 4.19	Often
2.60 - 3.39	Moderate frequency
1.80 - 2.59	Sometimes
1.00 - 1.79	Very rare

4.5.1 Frequency of problems

Table 4.22 represents the respondents' views toward the frequency of problems attributes. The attributes were measured by seven statements. According to the results in Table 4.22, it seems that problems of requirements elicitation in software development occurred more often than construction firms

Table 4.22: Descriptive analysis of frequency of problems attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
Requirements do not reflect the actual needs of the users/system stakeholders	1.83	.531	Sometimes	1.93	.365	Sometimes
Inconsistent requirements (i.e. requirements are stated in different and inconsistent ways in different parts of the document)	2.67	.479	Moderate frequency	2.47	.507	Sometimes
Incomplete requirements	2.17	.592	Sometimes	1.83	.531	Sometimes
Conflicting requirements (i.e. requirements come from different sources and they are not compatible to each other)	3.37	.556	Moderate frequency	2.77	.504	Moderate frequency
Misunderstood or misinterpreted requirements (i.e. different stakeholder communities use different jargon or different definitions for the same words).	1.63	.615	Very rare	1.37	.490	Very rare
Ambiguous (i.e. words are ambiguous)	1.30	.466	Very rare	1.23	.430	Very rare
Vague requirements (i.e. statements of the requirements fail to elicit an image or understanding of what is desired in the reader)	3.50	.572	Often	2.70	.535	Moderate frequency
Frequency of problems attributes	**2.35**	**.283**	**Sometimes**	**2.04**	**.173**	**Sometimes**

4.5.2 Problems of scope

Table 4.23 represents the respondents' attitudes toward problems of scope attributes. The problems of scope attributes were measured by two statements.

According to the results in Table 4.23, poor definition of system boundary was a serious problem for neither the software development firms nor the construction firms. Boundary conditions for the target system were well defined.

Table 4.23: Descriptive analysis of problems of scope attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
The boundary of the system is always ill-defined	2.53	.507	Disagree	2.33	.479	Disagree
Unnecessary design information is always given	3.90	.662	Agree	3.33	.711	Neutral/Not sure
Problems of scope attributes	**3.22**	**.387**	**Neutral/Not sure**	**2.83**	**.442**	**Neutral/Not sure**

4.5.3 Problems in understanding

Table 4.24 represents the respondents' attitudes toward problems in understanding attributes. The problems in understanding attributes were measured by seven statements.

The results show that problems in understanding in software development firms came from the fact that user and analyst speak different languages and that requirements are often vague and untestable, e.g., attributes of user friendly and robust.

Problems of users for having incomplete understanding of their needs and poor understanding of computer capabilities and limitations were insignificant in software development firms.

Table 4.24: Descriptive analysis of problems at understanding attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
Users have incomplete understanding of their needs	1.43	.568	Strongly disagree	1.40	.498	Strongly disagree
Users have poor understanding of computer capabilities and limitations	4.27	.521	Strongly agree	3.50	.572	Agree
Analysts have poor knowledge of problem domain	1.93	.521	Disagree	1.87	.346	Disagree
User and analyst speak different languages	3.83	.461	Agree	3.13	.507	Neutral/Not sure
There is difficulty of omitting "obvious" information	1.77	.568	Strongly disagree	1.67	.479	Strongly disagree
Conflicting views of different users	1.77	.430	Strongly disagree	1.63	.490	Strongly disagree
Requirements are often vague and untestable, e.g., "user friendly" and "robust"	3.43	.504	Agree	2.93	.521	Neutral/Not sure
Problems at understanding attributes	**2.63**	**.245**	**Neutral/Not sure**	**2.30**	**.236**	**Disagree**

4.5.4 Problems of volatility

Table 4.25 represents the respondents' attitudes toward problems of volatility. The problems of volatility attributes were measured only by one statement. According to the results in Table 4.25, it can be certainly said that during requirements elicitation of software development and construction companies, requirements evolved over time.

Table 4.25: Descriptive analysis of problems of volatility

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
Requirements evolve over time	4.27	.583	Strongly agree	3.93	.521	Agree
Problems of scope attributes	**4.27**	**.583**	**Strongly agree**	**3.93**	**.521**	**Agree**

4.5.5 Problems in requirements elicitation techniques

Table 4.26 represents the respondents' attitudes toward problems in requirements elicitation techniques attributes. The problems in requirements elicitation techniques attributes were measured by four statements. The table does not indicate any level of deep awareness of the effectiveness of techniques in either firm. They both indicated an awareness of the limitation of the techniques they currently use. This is somewhat of odds with their responses to the first three statements.

Table 4.26: Descriptive analysis of problems in requirements elicitation techniques attributes.

	Software development firms			Construction firms		
	Mean	SD	Rating	Mean	SD	Rating
The current requirements elicitation technique you use is inappropriate	2.93	.450	Neutral/Not sure	2.83	.461	Neutral/Not sure
The requirements elicitation techniques are dated	3.20	.551	Neutral/Not sure	3.03	.320	Neutral/Not sure
The requirements elicitation techniques are always fixed, regardless of the users' requirements and/or system developed	2.93	.254	Neutral/Not sure	3.03	.183	Neutral/Not sure
The capability and efficiency of the current requirements elicitation technique are limited	4.47	.571	Strongly agree	4.17	.379	Agree
Problems in requirements elicitation techniques attributes	**3.38**	**.260**	**Neutral/Not sure**	**3.27**	**.207**	**Neutral/Not sure**

CHAPTER 5
ANALYSIS

This chapter analyses the results of the survey according to the research aim and objectives. The results were obtained from the questionnaire survey on 60 companies who do construction and software development businesses in Thailand (30 each). Business information and demographic characteristics of the respondents from these firms are previously mentioned in chapter 4. The size of these companies was deemed large enough to be involved with projects requiring the use of requirement elicitation processes, and they used formalized process of project management and requirement engineering.

5.1 Requirement Elicitation in Thailand

Based on the results from section 4.3, the characteristics of requirements elicitation in Thailand was analyzed and described in terms of sources of requirements knowledge, techniques for requirements elicitation, and problem or difficulty in the process of requirements elicitation. The following subsections deal with these issues.

5.1.1 Sources of requirements knowledge

The respondents were asked to rank sources of requirements knowledge (customers or users' needs, domain information, existing system information, and other stakeholders' needs) that they acquired for system development (Rank "1" implying the most important source through to "4" implying the least important). This question was set up in order to determine the most important source of requirements knowledge used for software development and construction firms in Thailand.

5.1.1.1 Software development firms

According to the result in Table 4.10, the most important source of requirements knowledge for software development firms was customers or users' needs. This was in agreement with Saiedian and Dale (2000) that customers or buyers are the people responsible for contracting and paying for the target system. Their chief concerns include project schedule and budget, even to the point of compromising usability. Users are the individuals who ultimately will use the system developed. As such, they are most concerned with the usability, reliability, and availability of the system (Saiedian and Dale, 2000).

Customers or users' needs were previously reported both by researchers and practitioners to be the most important source of requirements knowledge and greatly affect requirement elicitation process. The results also agreed with the report of Przybilski (2006), the elicitation of end-user requirements is primary focus of the industry, where companies endeavor to discover and satisfy their customer's requirements. In order to gather requirements effectively it is necessary to identify the potential end-user and the targeted beneficiary of the effort. However the end-users often cannot express their needs (Orlikowski, 1993; Rossi and Tuunanen, 2004). Since the end-users are scattered and outside the traditional information systems development environment of a software development firm, it is not a trivial task to reach them.

The fact that customers or users' needs were the most important source of requirements knowledge for software development suggested that their state of requirement engineering is likely to be customer oriented. In general, there are market-driven and customer-specific software developments.

Domain information was found to be the second most important source of requirements knowledge for software development firms. Domain information is general information discussing the nature of the domain and the types of activities that are covered by the situation being considered (www2.umassd.edu, accessed in June 2009).

Other stakeholders' needs were found as the third rank of importance. This was in contrast to what Coulin and Zowghi (2005) reported. They found that stakeholders represented the most obvious source of determining requirements for a target system. Subject matter experts were used to supply detailed information about the problem and solution domains.

The least important source of requirements knowledge for software development firms was existing system information. Existing system information is information about systems that either will be replaced by the proposed system, or which the system must interact with (www2.umassd.edu). This also goes against the finding of Coulin and

Zowghi (2005), based on survey in Australia. They claimed that active systems and processes represented another important source for eliciting requirements, particularly when the project involved replacing an existing legacy system. Documentation on the current systems, processes, organization, and environment could provide a detailed foundation of requirements and supporting rationale.

One possible reason why software development firms in Thailand agreed that customers or users' needs were the most important source of requirements knowledge was the culture. Relationship-oriented behavior happens more commonly than work-oriented behavior in Thai society and its organization (Sorod, 1991). Moreover Thai people prefer to have stable social relationships and maintain surface harmony (Rohitratana, 1998). The characteristics of surface harmony are that a person is preferred to be smooth, kind, pleasant, conflict-free, non-assertive, polite, and humble (Rohitratana, 1998). Even after requirements were agreed to Thai analysts found themselves changing them to preserve good will of clients.

5.1.1.2 Construction firms

According to the result in Table 4.11, the most important source of requirements knowledge for construction firms too was customers or users' needs. Domain information was found to be the second important source of requirements knowledge. The least important source of requirements knowledge, it was other stakeholders' needs. Comparing with the software development firms, software development firms placed more importance on other stakeholders' needs than construction firms. This might relate to the fact that development software is a complicated process and requires several sources of knowledge from all involved stakeholders, particularly subject matter experts who could supply detailed information about the problem and solution domains.

5.1.2 Techniques for requirements elicitation

5.1.2.1 Software development firms

According to the result in Table 4.12, traditional techniques were the most frequently used technique to elicit requirements for software development firms. However the questionnaire survey in this study did not give more details on each traditional technique.

Group elicitation technique was also important for requirements elicitation in software development in Thailand as it received the second highest numbers of responses (see Table 4.12). This is in agreement with Coulin and Zowghi (2005) that group work is a well-established and often-used technique in requirements elicitation.

Table 4.12 showed that model-driven technique was the third range of techniques frequently used by software development firms in Thailand. Model-driven techniques usually provide a specific model of the type of information to be gathered, and use this model to drive the elicitation process (Tuunanen, 2003). Nuseibeh and Easterbrook (2000) have given examples of goal-based methods (e.g. van Lamsweerde and Letier 2000) and scenario-based methods (Maiden et al., 1999). This technique requires a thorough knowledge of the domain area of the system or a high level of knowledge of work practices which the end-users may not posses (Tuunanen, 2003). This might be a reason why this technique is unpopular amongst software development firms in Thailand. As Keretho (2008) has stated, one of the main challenge software development firms in Thailand confronted with was the unfamiliarity with domain area of the system. This might make model-driven technique unpopular in Thailand.

Prototyping technique was the fourth range of techniques frequently used in software development. Prototyping technique involves providing stakeholders with prototypes of the system to support the investigation of possible solutions is an effective way to gather detailed information and relevant feedback (Coulin and Zowghi, 2005). This technique provides the benefit of encouraging stakeholders, and more specifically the users, to play an active role in developing the requirements. It is very useful for developing new systems for entirely new applications. However the disadvantage is that it is generally expensive and it usually takes a long time to produce prototypes. This might be the reason why this technique is not popular for software development firms in Thailand.

Contextual technique was the fifth range of techniques frequently used in software development firms in Thailand according to Table 4.12. Contextual techniques, i.e. ethnographic techniques, involve the analyst overtly or covertly participating in the normal activities of the users over an extended period of time whilst collecting information on the operations being performed (Coulin and Zowghi, 2005). Generally ethnography is very expensive to perform and require significant skill and effort on the part of the analyst to interpret and understand the actions being performed. These reasons seem to make contextual technique not popular for eliciting requirements of software development firms in Thailand.

Table 4.12 also indicated that that cognitive techniques such as protocol analysis, laddering, card sorting, repertory grids were not used by software development in Thailand.

5.1.2.2 Construction firms

According to the result in Table 4.12, traditional techniques were the most frequently used techniques to elicit requirements for construction firms in Thailand. This was similar to the finding for software development firms.

Group elicitation technique came as the second rank for techniques frequently used by construction firm. This was also in agreement with software development firms. Both traditional and group elicitation techniques together accounted for 90% of total techniques used by the construction firms. It is therefore suggested that construction firms still relied on simple techniques for requirements elicitation since both traditional and group elicitation techniques do not use any advance technology such as modeling and prototyping. In addition the selection of these techniques by construction firms might be based on reasons such as:

(1) They are the only techniques that the analysts know
(2) They are the analyst's favorite techniques for all situations
(3) The analysts are following some explicit methodology, and that methodology prescribes a particular technique at the current time
(4) The analysts understand intuitively that the techniques are effective in the current circumstance (Hickey and Davis, 2002).

It is important to note that the choice of techniques to be employed depends on the specific context of the project and is often a critical factor in the success of the elicitation process (Nuseibah & Easterbrook, 2000). It is unlikely that a single technique can prescribe how to tackle the great variety of tasks and situations encountered by the systems analyst (Benyon and Skidmore, 1987). Inclusion of newer techniques by the construction firms may enhance the effectiveness of their requirements process.

Model-driven, prototyping and contextual techniques were not popular for construction firms in Thailand according to Table 4.12. Compared to software development firms, these techniques were used less by construction firms. This might relate to the advance technology required for these techniques. It is likely that software development firms have more available technological resources to perform model-driven, prototyping and contextual techniques than construction firms.

5.1.3 Problems or difficulties in the process of requirements elicitation

The result in Table 4.13 showed that construction firms seemed to have more problem or difficulties eliciting requirements than software development firms. This appears to be due to the fact that construction firms had a less systematic process for requirements elicitation. Construction firms' work mainly involved with supervising building. Most of the respondents from the construction firms were supervisors whose main role and responsibility was to supervise the building work while software development firms were more likely to be involved with projects related to system and software development.

5.2 State of requirements elicitation in Thailand

Based on the results from section 4.4, the state of requirements elicitation in Thailand was analyzed according to the end users/customers requirements, stakeholder requirements, domain-imposed (functional) requirements, management support/commitment, communication, and requirements documentation

5.2.1 End users/customers requirements

To evaluate the state of requirement elicitation in terms of end users/customers requirements, several statements were included in the survey given to the respondents to rate their attitudes (see Table 4.15).

5.2.1.1 Software development firms

According to the results in Table 4.15, in the process of requirement elicitations from end users/customers of software development firms, end users/customers requirements were considered to be the most important. Software/system engineers, managers and supervisors were concerned with the end users/customers preferences. However, the end users/customers requirements were always changed during the system development. This agreed with previous research conducted in Thailand by Thanasankit and Corbitt (2000) who found that Thai system analysts still had to change the requirements often according to the users/customers requirements:

"Interviewer: Do you ever tell your clients to stop changing requirements, this is enough?

Interviewee: No we can't. They are customers and if we do that, it will cut our relationship with them"

Moreover, the results in Table 4.15 indicate that in the process of requirement elicitations from end users/customers of software development firms, software/system engineers, managers and supervisors held regular discussions the end users/customers about their requirements. All the necessary requirements information was supplied from the end users/customers. This finding was surprising because it was opposite to previous study of Thanasankit and Corbitt (2000), who reported that some users tended to minimize their time available for providing system analyst with requirements because they believe that the new systems would threatening theirs and their colleagues' job security.

5.2.1.2 Construction firms

The end users/customers requirements for construction firms remained relatively unchanged during development, compared with the end users/customers requirements for software development firms. This might be due to the fact that the end users/customers requirements for system development in construction firms were not as complicated or vaguely defined as for software development. Typically development of software is a complicated process and requires a lot of requirements.

5.2.2 Other stakeholders' requirements

To evaluate the state of requirement elicitation in terms of other stakeholders requirements, several statements were constructed and given to the respondents to rate their attitudes (see Table 4.16).

5.2.2.1 Software development firms

The results in Table 4.16 indicate that, in the process of requirement elicitations from stakeholders of software development firms, software/system engineers, managers and supervisors seemed not to focus on stakeholders' requirements, unlike end users/customers requirements. This suggested that in the process of requirements elicitation by software development firms, other stakeholders' requirements were generally ignored. This is a serious issue because it can cause difficulties during system development. Many researches have shown that many large projects fail because of inadequate requirements (Boehm, 1981). Other stakeholders' requirements are important as well as end-users/customers requirements because they have a strong bearing on project outcomes (Coughlan et al., 2003). Thanasankit and Corbitt (2000) also found that there was a lack of involvement of technically knowledgeable stakeholders when defining the initial requirements in Thailand.

5.2.2.2 Construction firms

The results suggest that for construction firms, engineers, managers and supervisors were unlikely to have a chance to discuss with other stakeholders their requirements. Therefore other stakeholders' requirements were unlikely to be taken into consideration for system development.

5.2.3 Domain-imposed (functional) requirements

5.2.3.1 Software development firms

The results in Table 4.17 indicate that in the process of domain-imposed (functional) requirement elicitation, the domain-imposed (functional) requirements for system development were comprehensively obtained. Techniques, technology and software used for eliciting domain-imposed (functional) requirements in Thailand were updated. However, the complexity of software system complicated the requirements elicitation process. This was in agreement with Arayici and Aouad (2005) who state that software systems of today are characterized by increasing size, complexity, distribution, heterogeneity, and lifespan. They demand careful capture and modeling of requirements and architectural designs early on, before the low level system details begin to dominate the engineers' attention and significant resources are expanded for system construction (Grunbacher and Briggs, 2001).

5.2.3.2 Construction firms

Technology and software used for eliciting domain-imposed (functional) requirements of construction firms have not kept pace with state of the art. The perceived lack of need of such tools may be due to the fact that construction projects do not include complex software development and use. Tools such as CAD package may well be used for the design phases.

5.2.4 Management support/commitment

5.2.4.1 Software development firms

The results in Table 4.18 show that management of software development firms in Thailand exercised strong authority during requirement elicitations. This was in agreement with previous study by Thanasankit and Corbitt (2000) who found that the subordinates would follow their management although the requirements from the management might not be the correct ones. As a result Thai system analysts had to follow the consensus that emerged from the elicitation meeting and used them to design the relevant system. The social construction of subordinate agreement in Thai organizational contexts affects the quality and completeness of requirements gathering.

Effective requirements elicitation relies not only on senior management support but also individual project leaders' determination. Commitment does not simply mean giving approval (O'Hara, 2000). It includes, for example, provision of visible active support and encouragement. As good practice, management of software development firms in Thailand should support process improvement work by providing it with funding, staff and time (Kauppinen et al., 2004).

5.2.4.2 Construction firms

The results in Table 4.18 suggest that not all levels of management in construction firms provide the required support for the requirement elicitation. Management might not provide strong leadership for requirement elicitations and would not be willing to participate in meetings with end-users/stakeholders. Requirements elicitation practices may not be established by management as an integral part of the system development process and these practices would not be standardized across the projects done by the company. As a result of weak management support/commitment, training and resources for requirements elicitation implementation of construction firms appears to be lacking. Compared to software development firms, management support/commitment of construction firms for the requirement elicitations was much weaker.

5.2.5 Communication

Communication is important for the requirements engineering phase of software development projects. Various stakeholders must be able to communicate their requirements to the analysts, and the analysts need to be able to communicate the specifications they generate back to the stakeholders for validation (Al-Rawas and Easterbrook, 1996). Table 4.19 summarizes the investigation of this aspect in the chosen firm.

5.2.5.1 Software development firms

According to the results in Table 4.19, both verbal and well-written communications are used for information exchange between users and analysts.

5.2.5.2 Construction firms

The results indicate that in construction firms, both verbal and written communications were used regularly. Engineers, managers and supervisors always listened to what customers needed in order to preserve good will. However goals, objectives, tasks, procedures, constraints and timetables might not be regularly communicated between the firms and end-users/stakeholders. This was probably because there were intermediaries between the firms and the end users/customers. An example of an intermediary is the marketing and sales link (in which a salesperson serves as the intermediary) who defined corporate customer's goals and needs to designers and developers. Intermediaries often do not have a complete understanding of the customer's needs and, as a result, can intentionally or unintentionally filter and distort messages (Saiedian and Dale, 2000).

5.2.6 Requirements documentation

5.2.6.1 Software development firms

The results in Table 4.20 indicate that in software development firms, requirement documents might not be laid out in a highly readable manner. All of the necessary information, the services provides and functions performed by the system, and constraints in the design would not be discussed in the requirements documents. A glossary is not always included in the requirements documents. Finally reason why the system is required may be not clearly explained in requirements document. Based on the survey, requirement documentation seemed to be a serious issue in software development firms in Thailand.

5.2.6.2 Construction firms

The results in Table 4.20 suggest a better requirements documentation of construction firms compared to software development firms. This might relate to the fact that typically documentation for software development is complicated since it includes software technical words, programs, etc. The extent and goodness of documents, as indicated in Dahlstedt et al. (2004), varies between the companies.

5.3 Problems of requirements elicitation

Based on the results from section 4.5, problems of requirements elicitation in Thailand was analyzed in terms of frequency of problems, problems of scope, problems of understanding, problems of volatility and problems in requirements elicitation techniques.

5.3.1 Frequency of problems

5.3.1.1 Software development firms

According to the results in Table 4.22, the most often problem software development firms confront with was vague requirements. A vague requirement is one that is unclear or fuzzy rather than crisp (Carr, 2000). This corresponded to a previous study in the literature. Vague initial requirements problem was found to be the major problem in "Requirements problems in twelve software companies", the study of Hall et al. (2002).

Problems of inconsistent requirements (i.e. requirements are stated in different and inconsistent ways in different parts of the document) and conflicting requirements (i.e. requirements come from different sources and they are not compatible to each other) occurred moderately in software development firms. According to the theory, eliciting requirements for a proposed system inevitably involves the problem of handling problematic information about customers' needs, including inconsistency, vagueness, redundancy, or incompleteness (Mu et al., 2006). Inconsistency might be caused by the lack of cooperation between different parts of the organization as evidenced by Karlsson et al. (2007).

Inconsistent requirements mean that specific requirements are stated in different and inconsistent ways in different parts of the requirements document, particularly in a long or complex requirements document. Inconsistent requirements may come from the situation in which different people wrote the different sections, and they naturally come to the task with different viewpoints and writing styles (Carr, 2000). It is likely that inconsistent requirements found in software development firms in Thailand came from different levels of management in an organization having different perception about requirements for an information system (Thanasankit and Corbitt, 2000).

The survey showed that conflicting requirements often appeared during requirements elicitation process. This was in agreement with the literature that one of the major problems of this broadness of the range of stakeholders for large complex embedded systems is getting agreement between stakeholders (Nasr, 2005). In many cases the prioritization of requirements from different stakeholders groups is a source of much debate and dispute (Coulin and Zowghi, 2005). Often not all requirements are compatible with each other. Therefore analysts have to determine a suitable resolution through negotiation and compromise. This is the point where trade-offs are made (Carr, 2000). The analysts have to prioritize the requirements first in order to know which is more valuable and which is less valuable to the overall system. Moreover the analysts have to be sensitive to all the political and organizational aspects of the project when mediating discussions related to the target system (Coulin and Zowghi, 2005).

In requirements elicitation of software development firms, incomplete requirements sometimes appeared as well as requirements that did not reflect the actual needs of the users/system stakeholders. According to Gause and Weinberg (1989), human beings are typically good at pattern recognition but their abilities often do not extend to determining missing elements in a pattern. When investigating a set of requirements any one individual is likely to overlook many requirements through unfamiliarity (Carr, 2000). Therefore it is suggested to employ a diverse group of reviewers to help ferret out missing requirements. Requirement analysis techniques adverted in systems engineering will also be of use.

Requirements that did not reflect the actual needs of the users came from the fact that both users and engineers find it easier to think in concrete, rather than abstract, terms (Carr, 2000). Therefore they will often report a design solution as part of their requirements document. The requirement statement should address a need, not an implementation solution.

For software development firms in Thailand, problems of ambiguous requirements and misunderstood/misinterpreted requirements rarely occurred.

5.3.1.2 Construction firms

According to the results in Table 4.22, the problems construction firms moderately confronted with were vague requirements and conflicting requirements.

Problem of vague requirements was found to be prevalent for both software development and construction firms. Therefore both software development and construction firms in Thailand should take the problem of vague requirements into consideration seriously when they carry out any system and/or software development, and use appropriate classification techniques to address there.

Conflicting requirements were also found important in construction firms in Thailand. This was in agreement with Arayici and Aouad (2005) who have reported that because the construction industry was a complicated one, this complexity would be reflected in the requirements specification. Besides, it is quite probable that there would be conflicts between the needs of different stakeholders which would add an extra layer to the complexity of the requirements specification.

The problems that sometimes occurred in the process of requirements elicitation of construction firms in Thailand include inconsistent requirements, incomplete requirements, and that requirements did not reflect the actual needs of the users/system stakeholders. This was similar to the case of software development firms.

For construction firms, problems of ambiguous requirements and misunderstood/ misinterpreted requirements rarely appeared. This was same as software development firms.

5.3.2 Problems of scope

To evaluate the problems of scope occurring in requirements elicitation process, two statements were assigned and given to the respondents to rate their attitudes (see Table 4.23).

5.3.2.1 Software development firms

The results in Table 4.23 suggest that ill-defined boundary of the system was not a serious problem for software development firms. Boundary conditions for the target system were well defined. This might relate to the appropriate elicitation strategies the firms used because well-defined boundary of the system can be achieved with appropriate elicitation strategies (Christel and Kang, 1992). Elicitation techniques need to be broad enough but still should focus on the creation of requirements, not design activities, in order to adequately address users' concerns and not just developers' needs. Elicitation strategies that produce requirements in the form of high level designs might not be verifiable by the users since they cannot adequately understand the design language.

As good practice, it is recommended that software development firms in Thailand should consider the following factors in the process of requirements elicitation (Christel and Kang, 1992):

- Organization factors: these include submitters of input to the target system, users of the target system's output, and ways in which the target system will change the organization's means of doing business

- Environmental factors: these include hardware and software constraints imposed on a target system, the maturity of the target system's domain, the certainty of the target system's interfaces to the larger system, and the target system's role within a larger system

- Project factors: these include the attributes (e.g. management style, management hierarchy, domain experience, computer experience) of the different stakeholder communities, such as the end users, sponsors, developers, and requirements analysts, and the constraints imposed by the people involved in the elicitation process, e.g., managerial constraints concerning cost, time, and desired quality in the target system.

5.3.2.2 Construction firms

According to the results in Table 4.23, in the process of requirements elicitation of construction firms, unnecessary design was not usually included. The lower score of this statement compared to software development firms also suggested that software development firms obtain more unnecessary design information than construction firms. This might be done to the fact that a lot of information is sought for a software development. The more we search for information, the more we found unnecessary design information (Nuseibeh and Easterbrook, 2000).

Ill-defined boundary of the system was not a serious problem for construction firms according to the survey. This is in agreement with software development firms.

5.3.3 Problems of understanding

To evaluate the problems at understanding during requirements elicitation process, seven statements were assigned and given to the respondents to rate their attitudes (see Table 4.24).

5.3.3.1 Software development firms

According to the results in Table 4.24, problems of understanding in software development firms came from the fact that user and analyst speak different languages and that requirements are often vague and untestable, e.g., using terms such as user friendly and robust. Using natural language, one word may have different meanings to different people (Karlsson et al., 2007).

Christel and Kang (1992) have reported that the language used to express the requirements among stakeholder communities (customers/sponsors, users, developers, quality assurance teams, and requirements analysts) might be too formal or too informal to meet the needs of each of the group. As a result, different languages might be used by diverse communities. Requirements are generated and interpreted by people with different experience levels and backgrounds. The form and language in which the requirements are expressed can affect understanding. This can lead to ambiguous, inconsistent, incorrect and incomplete requirements.

According to Table 4.24, problems of users for having incomplete understanding of their needs and poor understanding of computer capabilities and limitations were insignificant in software development firms. This disagreed with a previous study of Rajagopal et al. (2005) that, requirements elicitation process should be improved by training the non-technical stakeholders (primarily the users) in the capabilities and limitations of computer hardware, software, and of software developers.

The results in Table 4.24 also indicated that different users did not have conflicting views. In terms of analysts, they had enough knowledge of problem domain and they did not have difficulty of omitting obvious information. However, Thanasankit and Corbitt (2000) have pointed out that although Thai system analysts had enough knowledge, Thai culture still affected the requirements elicitation process. Young system analysts often followed the ways that senior system analysts undertook their work.

5.3.3.2 Construction firms

The results in Table 4.24 indicated that problems of understanding in construction firms were similar to software development firms.

5.3.4 Problems of volatility

To evaluate the problems of volatility, a statement was assigned and given to the respondents to rate their attitudes (see Table 4.25).

5.3.4.1 Software development firms

According to the results in Table 4.25, the respondents from software development firms strongly agreed with the statement that "Requirements evolve over time".

A number of different reasons have been given for causing requirements change, e.g. fluctuation of the market, problems occurring during coding and reviews, and resource constraint change (Karlsson et al., 2007). According to Sage and Palmer (1990), a major cause of requirements volatility is that "user needs evolve over time". The requirements engineering process of elicit, specify, and validate should not be executed only once during system development, but rather should be returned to so that the requirements can reflect the new knowledge gained during specification, validation, and subsequent activities (Christel and Kang, 1992). A requirements engineering methodology should be iterative in nature, "so that solutions can be reworked in the light of increased knowledge" (Macaulay et al., 1990, p. 102). Therefore, software development firms and construction firms in Thailand should consider if they could gain knowledge during the evolvement, otherwise it would really waste of time and budget.

The evolution of requirements over time might be associated with Thai culture in which decision-making is typically confined to high level of management, not a team's decision. For example, further changes to the requirements can be done although clients have approved the requirements. Thai system analysts still have to change the requirements and gather more to preserve good will and preserve their relationship with the clients. Often this is associated with issues raised by those at the top management who see a need to show they have had some input (Thanasankit and Corbitt, 2000).

5.3.4.2 Construction firms

According to the results in Table 4.25, the respondents from construction firms agreed with the statement that "Requirements evolve over time". This also happened in the case of software development firms. The evolution of requirements over time of both software development and construction firms was in agreement with the theory because at each stage of the project, addition information is derived and documented.

5.3.5 Problems in requirements elicitation techniques

To evaluate the problems in requirements elicitation techniques used in requirements elicitation process, four statements were assigned and given to the respondents to rate their attitudes (see Table 4.25).

5.3.5.1 Software development firms

According to the results in Table 4.25, it can be interpreted that current techniques used for requirements elicitation of software development firms in Thailand had limited capability and efficiency. Such current techniques included traditional techniques and group elicitation techniques. It is recommended for software development firms in Thailand to keep traditional and group elicitation techniques but at the same time consider new methods and techniques as supplements. Moreover, the current requirements elicitation techniques used in software development firms might not be appropriate. Therefore, the firms needed to evaluate the appropriateness of the techniques they currently used in order to solve this problem. Moreover realizing the performance of the technique could lead to improvement of the techniques or changing the techniques to the more efficient ones.

Another problem of software development firm is that the requirements elicitation techniques might be always fixed regardless of the users' requirements and/or system developed. This might relate to the fact that system analysts in Thailand always follow their senior analysts who may be used to some preferred techniques (Thanasankit and Corbitt, 2000).

5.3.5.2 Construction firms

According to the results in Table 4.25, construction firms are similar to software development firms. Therefore, similar recommendation can be made.

5.4 Comparison of problems of requirements elicitation

When comparing the attitude scores towards problems of scope, problems at understanding, problems of volatility, and problems in requirements elicitation techniques of software development firms, problems of volatility were rated with the highest score, followed by problems of requirement elicitation techniques, problems of scope, and problems at understanding. This suggested that software development firms in Thailand should focus on addressing the problems of volatility.

For construction firms, similar result was obtained. Problems of volatility were rated with the highest score, followed by problems of requirement elicitation techniques, problems of scope, and problems at understanding. This suggested that construction firms should also concentrate on addressing the problems of volatility.

5.5 Comparison of requirement elicitation state in Thailand with good practice

The results from questionnaire survey in section 4 were classified, compiled and compared to the good practices for requirements elicitation as summarized in this section. Good practices were taken from academic text books including:

- Requirements engineering: a good practice guide, written by Ian Sommerville and Pete Sawyer (1997);
- Requirements engineering: processes and techniques, written by Gerald Kotonya and Ian Sommerville (1998);
- Software requirements, written by Karl E. Wiegers (1999);
- The requirements engineering handbook, written by Ralph R. Young (2004).

These good practices are also related to the principle steps of requirements elicitation which consist of (Humphrey, 1999):

- Identification of stakeholders of the project
- Recording requirements sources and project glossary
- Defining system boundaries
- Recording documentation of the rationale for each requirement
- Using prototyping technique for uncertain requirement
- Using scenarios to elicit requirement
- Selecting the appropriate practices, methods, and techniques for gathering requirements
- Taking time to educate the customers and consider their preference
- Providing requirements-related training for project participation
- Using group elicitation technique to foster stakeholder agreement and buy in

Table 5.1: Arbitrary level for attitudes towards states of requirements elicitation

Arbitrary level	Descriptive rating
4.20 - 5.00	Strongly agree
3.40 - 4.19	Agree
2.60 - 3.39	Neutral/Not sure
1.80 - 2.59	Disagree
1.00 - 1.79	Strongly disagree

5.5.1 Identification of stakeholders of the project

Good practice of requirements elicitation requires identification of stakeholders of the project. According to Suzanne (2000), it is a good practice to identify stakeholders of the project at the first stage of a project. The reason stakeholders is necessary is because it is much better opportunity that people will be involved to the project. Also, they are the composition of the project team which has a strong bearing on project outcomes (Coughlan et al., 2003). Therefore, knowledge and better understanding about the project would be contributed.

Table 5.2 represents the respondents' attitudes toward identification of stakeholders of the project attribute. This attribute was measured only by one statement. The average scores obtained from both software development (2.90) and construction firms (3.07) in Thailand were moderate, suggesting that good practice in term of identification of stakeholders of the project in Thailand might not be established yet. Identification of all stakeholders is necessary for any system development. A stakeholder is anyone who has an interest in the project and anyone who will be touched by the system (Young, 2004). Both software development and construction firms need to carefully think of all stakeholders when they come to the process of requirements elicitation. Those stakeholders include customers (those who are paying for the work), users (people who will actually use the system), advisors (such as legal experts or regulators who have relevant information about the requirements), project groups that are involved in developing the system, (such as system engineering, software engineering, QA, CM, project control, documentation, training, testing, and so forth). Several techniques can be used to identify stakeholders such as stakeholder analysis template (Sommerville and Sawyer, 1997).

Construction firms in Thailand better identified the stakeholders of the project than software development firms. This might relates to the fact that the system development of construction firms is typically less complicated than that of software development firms. Moreover, large firms had a better identification of the stakeholders than medium-sized firms.

Table 5.2: The descriptive analysis of identification of stakeholders of the project

	Software development firms			Construction firms		
	Mean			Mean		
	Medium size	Large size	Average	Medium size	Large size	Average
You take other stakeholders requirements into consideration for the system development as well as end users/customers' requirements	2.80	3.00	2.90	2.90	3.20	3.07
Identification of stakeholders of the project attribute	2.80	3.00	2.90	2.90	3.20	3.07
Rating	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure

5.5.2 Recording requirements sources and project glossary

A glossary is basically a list of terms in particular domain of knowledge with the definitions. Ralph R. Young (2003) recommended that a project glossary should be developed in each project to avoid the misunderstanding of definitions in project documentation.

Table 5.3 represents the respondents' attitudes toward recording requirements sources and project glossary attributes. These were measured by 2 statements. The average scores obtained from both software development (2.78) and construction firms (2.62) in Thailand on this issue were moderate, suggesting that good practice in term of requirements sources and project glossary recording in Thailand might not be established yet.

Although, it is not an easy task to record all the requirements sources for the development of a complex system. The requirements sources and project glossary is still needed as they could help project documentation staying away from ambiguity. Leffingwell and Widrig (2000) have recommended that complex systems entail requirements specification for each sub-system, and non-trivial applications, requirements must be captured and recorded in a document database, model or tool.

Defining project glossary is very important since technical people tend to have strong opinions about definitions of particular word and they tend to resist moving forward when they do not know or understand an acronym that is begin used (Young, 2004). This will cause too much discussion, confusion, frustration and delays of the project. It is therefore strongly recommended for both software development and construction firms in Thailand to develop a project glossary and a project list of acronyms as early as possible to avoid the delay of the system development.

Software development firms in Thailand better recorded requirements sources and project glossary than construction firms. This might relates to that the more complication of system development in software companies have pushed them to carefully record the sources of requirements and the project glossary. Moreover, the large-sized firms better recorded the requirements sources than the medium-sized firms.

Table 5.3: The descriptive analysis of recording requirements sources and project glossary

	Software development firms			Construction firms		
	Mean			Mean		
	Medium size	Large size	Average	Medium size	Large size	Average
All of the necessary information is provided in requirements documents, even if only by reference	2.80	3.07	2.93	2.93	3.20	3.07
Glossary is always included in the requirements documents	2.53	2.73	2.63	2.00	2.33	2.17
Recording requirements sources and project glossary attributes	2.67	2.90	2.78	2.47	2.77	2.62
Rating	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure	Disagree	Neutral/ Not sure	Neutral/ Not sure

5.5.3 Defining system boundaries

According to Mochal (2006), it is an important as a good practice that system boundaries and scope should be defined and documented because it helps all stakeholders understand the planned system. It should be done in the early stage of requirement elicitation activities.

Table 5.4 represents the respondents' attitudes toward defining system boundaries attributes. The defining system boundaries attributes were measured only by one statement. The results showed that both software development firms and construction firms overall disagreed on "the boundary of the system is always ill-defined" statement. This mean ill-defined system boundary is not a problem for both software development firms and construction firms. Therefore, the problems of requirement elicitation in Thailand might be other aspects, not ill-defined boundary system.

Table 5.4: The descriptive analysis of defining system boundaries

	Software development firms			Construction firms		
	Mean			Mean		
	Medium size	Large size	Average	Medium size	Large size	Average
The boundary of the system is always ill-defined	2.53	2.53	2.53	2.67	2.40	2.33
Defining system boundaries attributes	2.53	2.53	2.53	2.67	2.40	2.33
Rating	Disagree	Disagree	Disagree	Neutral/ Not sure	Disagree	Disagree

5.5.4 Recording documentation of the rationale for each requirement

Good practice requires recording requirements rationale in order to improve requirements understanding. Table 5.5 represents the respondents' attitudes toward recording documentation of the rationale for each requirement attribute that were measured by 4 statements. The average scores obtained from both software development (2.94) and construction firms (2.60) in Thailand were moderate, suggesting that good practice in term of recording requirements rationale might not be established yet. This, if it indicate that rationale is not recorded, may lead to interpretation problems later. This in turn, will result in problem of implementation and unification.

Recording documentation of the rationale for each requirement is important since it can reduce cost, time and effort significantly. The rational behind each requirement is necessary and should not be ignored because it explains the situation of the project and it removes ambiguity which may occur (R. Ian Faulconbridge and Michael J. Ryan, 2002). Moreover, Finkelstein (1994) states that in most system development processes, more than 70% of costs are in rework and half the effort in these activities is about understanding the system in order to make effective corrections and enhancements. To achieve this understanding it is necessary to know what decisions were considered, assumptions made and alternative solutions rejected. Without recording, this information can be lost with time and the staff turnover. Therefore it is essential to keep a record of the rationale underpinning the products of requirements engineering (Finkelstein, 1994). Moreover, Young (2004) has recommended to track the source of any rationale that is accepted during the requirements elicitation process in order to easily defend the decision or to review it if assumptions change or are proven false.

When comparing the performance of recording requirements rationale between software development firms and construction firms in Thailand, it was found that construction firms had poorer recording requirements rationale and the medium-sized firms had poorer recording requirements rationale than the large-sized firms.

Table 5.5: The descriptive analysis of recording documentation of the rationale for each requirement

	Software development firms			Construction firms		
	Mean			Mean		
	Medium size	Large size	Average	Medium size	Large size	Average
Requirement documents are laid out in a highly readable manner.	3.20	3.53	3.37	3.33	3.80	3.57
Requirements documents clearly explain why the system is required	2.80	2.93	2.87	2.33	2.53	2.43
All of the services provided and functions performed by the system are discussed in the requirements documents	2.53	2.67	2.60	2.00	2.13	2.07
All constraints in the design are discussed in the requirements documents	2.80	3.00	2.90	2.33	2.33	2.33
Recording documentation of the rationale for each requirement attributes	2.83	3.03	2.94	2.50	2.70	2.60
Rating	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure	Disagree	Neutral/ Not sure	Neutral/ Not sure

5.5.5 Using prototyping technique for uncertain requirement

The result in Table 5.6 shows that only 3 out of 30 software development firms and 1 out of 30 construction firms in Thailand have used prototyping technique for uncertain requirements (10.0% and 3.33% correspondingly). This ratio is very low, indicating prototyping technique is hardly used for addressing uncertain requirements for system development in Thailand. Moreover, only large-sized firms have used this technique. This might due to the prototyping technique is generally expensive. This might be the reason why this technique is not popular.

According to Davis (1992), prototyping technique is an important technique for requirements elicitation where customers are uncertain of exactly what their needs or where early feedback from stakeholders is needed. It is a technique commonly used when the requirements are not clear. It is process of developing a rough replica of the proposed system so as to get a clear understanding of the requirements (Austin, 2007). It is the construction of an executable system model to "enhance understanding of the problem and identify appropriate and feasible external behaviors for possible solutions" (Hsia et al., 1993).

Uncertain requirements can come from customers who are uncertain of exactly what they want or need. System engineers therefore might get only a "wish list" of what the customers would like to see, with few details or without being clear as to whether the list is complete (Pfleeger and Atlee, 2005). Moreover, it is easy for many people to criticize, in detail, an existing product than to image, in detail, a new product. As such, one way that we can elicit details is to create a prototype of the proposed system and to solicit feedback from potential users about what aspects they would like to see improved (Pfleeger and Atlee, 2005). Using prototyping technique can help in determining whether the customer's problem has a feasible solution, or exploring options for optimizing quality requirements.

Table 5.6: The descriptive analysis of using prototyping technique for uncertain requirement

	Software development firms			Construction firms		
	Numbers of firms			Numbers of firms		
	Medium size	Large size	Total	Medium size	Large size	Total
Most frequently used Prototyping technique for requirements elicitation	-	3 (20.0%)	3 (10.0%)	-	1 (6.67%)	1 (3.33%)
Using prototyping technique for uncertain requirement attributes	-	3 (20.0%)	3 (10.0%)	-	1 (6.67%)	1 (3.33%)

5.5.6 Using scenarios to elicit requirement

The result in Table 5.7 shows that only 5 out of 30 software development firms and 1 out of 30 construction firms in Thailand have used scenario to elicit requirements (16.67% and 3.33% correspondingly). These low numbers suggest scenario is hardly used to elicit requirements in Thailand. Moreover, only large-sized firms have used this technique. This is in agreement with Cox (2008) who found that the requirements discovery phase is less served by scenario approaches than other phases.

A scenario is one of several techniques used for requirements elicitation. Scenarios are defined as "informal narrative descriptions…stories about human activity", and are used "to conduct analysis and design in a vocabulary that permits end-user participation" (Carroll, 2000). Scenarios are often particularly defined situations including a particular setting, a central character or "agent" and a plot, consisting of a sequence of actions and events (Carroll, 2000).

Scenarios are useful for requirements gathering because they permit people to discuss situations without reference to specific technologies. This is particularly valuable when working with non-professional people as their frequent lack of knowledge about technical language and different technologies can be often prove a barrier in requirements gathering (Eisma et al., 2003). Many people in Thailand are likely to lack of technical language and be unfamiliar with technologies as previously found by Thanasankit and Corbitt (2000) that there was a lack of involvement of technically knowledgeable stakeholders when defining the requirements in Thailand. Therefore, using scenarios might be a good option for requirements elicitation in Thailand.

Table 5.7: The descriptive analysis of using scenarios to elicit requirement

	Software development firms			Construction firms		
	Numbers of firms			Numbers of firms		
	Medium size	Large size	Total	Medium size	Large size	Total
Most frequently used model-driven technique for requirements elicitation (i.e. scenario-based methods)	-	5 (33.0%)	5 (16.67%)	-	1 (6.67%)	1 (3.33%)
Using scenarios to elicit requirement attributes	-	5 (33.0%)	5 (16.67%)	-	1 (6.67%)	1 (3.33%)

5.5.7 Selecting the appropriate practices, methods, and techniques for gathering requirements

Good practice of requirements elicitation requires the selection of appropriate practices, methods and techniques for gathering requirements. The average scores obtained from both software development (2.77) and construction firms (2.67) in Thailand were moderate, suggesting that good practice in term of the selection of appropriate practices, methods and techniques for gathering requirements in Thailand might not be established yet.

Selecting appropriate techniques for gathering requirements is not easy. Since there is no universal technique that can be used for all situations and each situation is unique, system developers need to identify which technique (or techniques) best suits a specific project. Of course many techniques are available in the literature and each one has advantages and disadvantages differently.

The non-established good practices in terms of the selection of appropriate practices, methods and techniques for gathering requirements of software development and construction firms in Thailand might relate to the fact that those companies do not have much experience in software and/or system development. In the study of Hickey and Davis (2003) it was found that less experienced analysts often select a technique based on one of two reasons: (i) it is the only one they know, or (ii) they think that a technique that worked well last time must surely be appropriate this time. As a result of such inappropriate selection, less experienced analysts seem to be less successful in uncovering the user needs than extensive experienced analysts.

Table 5.8: The descriptive analysis of selecting the appropriate practices, methods, and techniques for gathering requirements

	Software development firms			Construction firms		
	Medium size	Large size	Average	Medium size	Large size	Average
Techniques, technology and software used for eliciting domain-imposed (functional) requirements are updated	3.20	3.60	3.40	2.80	3.00	2.90
The current requirements elicitation technique you use is appropriate	2.40	2.93	2.67	3.13	3.20	3.17
The requirements elicitation techniques are up-to-dated	2.93	3.33	3.13	2.93	3.00	2.97
The requirements elicitation techniques are not always fixed, regardless of the users' requirements and/or system developed	3.07	3.07	3.07	3.13	2.60	2.87
The capability and efficiency of the current requirements elicitation technique are not limited	1.67	1.53	1.60	1.20	1.67	1.44
Selecting the appropriate practices, methods, and techniques for gathering requirements attributes	2.65	2.89	2.77	2.64	2.69	2.67
Rating	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure	Neutral/ Not sure

5.5.8 Taking time to educate the customers and consider their preference

Good practice of requirements elicitation requires time taken to educate the customers and consider their preference. The average scores obtained from both software development (4.01) and construction firms (3.62) in Thailand were high, suggesting that good practice in term of time taken to educate the customers and consider their preference in Thailand is already established. Moreover, software development firms seem to take more time to educate the customers and consider their preference than construction firms.

Software success depends on developing a collaborative partnership between software developers and their customers. Too often, though, the customer-developer relationship becomes strained or even adversarial. Problems arise partly because people don't share a clear understanding of what requirements are and who the customers are.

Table 5.9: The descriptive analysis of taking time to educate the customers and consider their preference

	Software development firms			Construction firms		
	Medium size	Large size	Average	Medium size	Large size	Average
You regularly discuss with the end users/customers about their requirements.	3.33	3.67	3.50	3.67	3.80	3.73
You completely obtain all the necessary requirements information from the end users/customers.	4.33	4.53	4.43	2.93	3.13	3.03
When you come to the process of requirements elicitation from end users/customers, you are concerned with what they do not want as well as what they want.	4.67	4.60	4.63	4.40	4.53	4.47
It can be said that in the process of requirements elicitation, end users/customers requirements are the most important	4.53	4.67	4.60	4.40	4.53	4.47
Management is willing to participate in meetings with end-users/stakeholders	3.47	3.67	3.57	2.80	2.87	2.83
You always communicate goals, objectives, tasks, procedures, constraints, timetables to end-users/stakeholders regularly	3.27	3.67	3.47	2.93	3.20	3.07
You are always listening to what customers need in order to impress them	3.80	4.00	3.90	3.53	4.00	3.77
Taking time to educate the customers and consider their preference attributes	3.91	4.11	4.01	3.52	3.72	3.62
Rating	Agree	Agree	Agree	Agree	Agree	Agree

5.5.9 Providing requirements-related training for project participation

Table 5.10 represents the respondents' attitudes toward providing requirements-related training for project participation attribute. Good practice of requirements elicitation requires the provision of requirements-related training for all project participants. The average scores obtained from both software development (3.37) and construction firms (2.60) in Thailand were moderate, suggesting that good practice in term of providing requirements-related training for project participants in Thailand might not be established yet. Software development firms seem to be more focus on this issue than construction firms due to the higher score.

Industry systems engineering and requirements trainer Robert Halligan believes that the number one problem in requirements engineering is that project managers fail to require experienced requirements analysts (RAs), and that RAs are not sufficiently trained and/or experienced to perform their roles effectively (Whitten, 2005). The fields of training by Young (2004) were described in literature review chapter.

Although the requirements-related training is set, the project-involved participants might refuse to join the training with the reasons such as they don't need it, they are very busy with their own work. It is a challenge for the software development and construction firms in Thailand to make them realize the value of investment in the requirements process and related information. This is particularly true for construction firms since they obtained lower score than software development firms.

Table 5.10: The descriptive analysis of providing requirements-related training for project participation

	Software development firms			Construction firms		
	Medium size	Large size	Average	Medium size	Large size	Average
Management is committed to provide training and resources for requirements elicitation implementation	3.13	3.60	3.37	2.47	2.73	2.60
Providing requirements-related training for project participation attributes	3.13	3.60	3.37	2.47	2.73	2.60
Rating	Neutral/ Not sure	Agree	Neutral/ Not sure	Disagree	Neutral/ Not sure	Neutral/ Not sure

5.5.10 Using group elicitation technique to foster stakeholder agreement and buy in

The result in Table 5.11 shows that 9 out of 30 software development firms (30%) in Thailand have used group elicitation technique to foster stakeholder agreement and buy in. Large-sized firms have used the group elicitation technique more than medium-sized firm. For construction firms, the group elicitation technique has been used more. 12 out of 30 construction firms (40%) in Thailand have used group elicitation technique to foster stakeholder agreement and buy in. Similar to software development firms, large-sized construction firms have used the group elicitation technique more than medium-sized firm. Less than half of the survey firms using the group elicitation technique suggests that good practice of using group elicitation technique to foster stakeholder agreement and buy in is somewhat established.

Group elicitation technique is particularly effective since it would appear that groups are able to deal with complex tasks such as requirements elicitation better than individuals because they have a wider range of skills and abilities to draw from (Coulin, 2007). Group elicitation techniques aim to exploit team dynamics (Nuseibeh & Easterbrook 2000). Group work techniques are also beneficial because they involve the users, commit the customers, and promote discussion, collaboration, idea generation, solution finding, and decision making (Coulin, 2007). Moreover group techniques are naturally very important to the requirements elicitation process because software development is inherently a group effort (Palmer & Fields 1992).

However, for effective requirement elicitation process, it is suggested for firms to use group elicitation technique with other techniques. No one technique in isolation is able to capture all the requirements completely (Maiden and Rugg, 1996). This is particularly true for software development firms because more than one technique is required to elicit all the actual requirements for complex software-based systems

Table 5.11: The descriptive analysis of using group elicitation technique to foster stakeholder agreement and buy in

	Software development firms			Construction firms		
	Numbers of firms			Numbers of firms		
	Medium size	Large size	Total	Medium size	Large size	Total
Most frequently used group elicitation techniques for requirements elicitation	2 (13.0%)	7 (46.67%)	9 (30.0%)	3 (20.0%)	9 (60.0%)	12 (40.0%)
Using group elicitation technique to foster stakeholder agreement and buy in attributes	2 (13.0%)	7 (46.67%)	9 (30.0%)	3 (20.0%)	9 (60.0%)	12 (40.0%)

CHAPTER 6
CONCLUSIONS AND RECOMMENDATION

6.1 Summary of thesis and conclusion

In this research, the concept of requirement elicitation has been investigated based on literature review of requirement engineering and surveys. The major objective of this research was to investigate the state of requirements elicitation practice in Thailand. Requirement constraints and compliance to requirements in a system are also considered.

To achieve the research objective, several questions have been used as a guide as follows:

1. What is the current state of requirement elicitation practice in Thailand?
2. How effective is it?
3. What are the main problems that occur during the requirement elicitation process?
4. Why do they occur?
5. How may they be addressed or solved?
6. How does the practice compare with good practice with theory?

It was discovered from the survey that most software development and construction firms in Thailand have used requirements engineering and requirements elicitation. The study showed that most software development and construction firms in Thailand encountered problems or difficulties in the process of requirements elicitation. These problems included frequency of problems, problems of scope, problems at understanding, problems of requirements volatility and problems in requirement elicitation technique. These problems need to be addressed.

It showed that the current requirement elicitation in Thailand is not good and it needs to be improved. Construction firms seemed to have more problems than software development firms.

The most often problem both types of firms confronted with was vague requirements. Therefore both should take the problem of vague requirements into consideration seriously when they carry out any system development.
The serious issue for software development firms is poor requirement documentation. This is because they fail to discuss all of the necessary information, services provides and functions performed by system, and constraints in the design. Also, a glossary is not always included in requirements documents.

Another problem is that stakeholders' requirements were not taken into consideration seriously. Therefore software development and construction firms in Thailand need to be more concerned and emphasize stakeholders' role in the process. Suitable techniques should be used to aide this process.

The result also shows that the current requirements elicitation techniques were not appropriate. Therefore it is suggested for software development and construction firms in Thailand to evaluate the efficiency and effectiveness of the technique they currently use for requirement elicitation. And make changes in their process as needed.

In comparison with best practices, the results showed that in Thailand, good practice in terms of identification of consult system and stakeholders of the project, requirements sources and project glossary recording, recording requirements rationale, selection of appropriate practices, methods and techniques for gathering requirements, providing requirements-related training for project participants, were not be established yet.

The good practice in term of indicating prototyping technique for addressing uncertain requirements, and using scenarios is hardly used to elicit requirements for the candidate firms in Thailand.

The good practice of using group elicitation technique to foster stakeholder agreement and buy in is somewhat established. Moreover, they do not have problems with ill-defined boundary system and the good practice in term of time taken to educate the customers and consider their preference in Thailand is already established.

Finally, as indicated in Chapter 5, the research questions were answered, using either the primary or secondary data sources, and the research objectives were therefore achieved.

6.2 Recommendations

I believe that the three recommendations that will yield the largest benefit to the candidate firms are;

- In the process of requirement elicitations of software development and construction firms in Thailand, stakeholders requirements were not taken into consideration seriously, other than customers/end-users needs. Stakeholders' requirements are important because stakeholders are the composition of the project team which has a strong bearing on project outcomes (Coughlan et al., 2003). For example, stakeholders are needed to participate in requirements elicitation techniques (Coulin and Zowghi, 2005) and good communication between stakeholders and analysts is required. A recommendation from this research is that software development and construction firms in Thailand need to be more proactive and emphasize stakeholders' role and involvement in requirement elicitation.

- Problem of vague requirements (i.e. statements of the requirements fail to elicit an image or understanding of what is desired in the reader) are found in both software development and construction firms. Therefore both should take the problem of vague requirements into consideration seriously when they carry out any system development. Requirement analysis techniques should be employed to facilitate this. Stakeholder involvement indicated above, together with techniques another prototyping will enhance this aspect.

- The results about requirement elicitation techniques from both software development and construction firms implied that the current requirements elicitation techniques may be inappropriate. Engineers, managers and supervisors of the firms might not know whether the requirements elicitation techniques they used are dated or not. Therefore, it is suggested to evaluate the effectiveness of the techniques they currently use for requirement elicitation. Realizing the performance of the technique leads to improvement of the techniques or changing the techniques to more effective ones. Moreover, combinations of techniques might be more appropriate than a single technique.

6.3 Future Research Direction

Future research may investigate specific elicitation techniques used in Thailand in more detail. Techniques are crucial to the success of requirements elicitation process because the aim of all techniques is to find out what users really need. It is added that other stakeholders' requirements are also important. Appropriate techniques have to concern with other stakeholders' requirements as well as users' requirements.

The questionnaire survey in this study did not give more details on each traditional technique. It is therefore suggested that further study be carried out to investigate the use of traditional techniques in Thailand.

Lastly, the analysis in this research is element by element and has some cross element correlation of information or data. Therefore, it would be a potential resource for further research about cross elements.

REFERENCE

Abran, A., Moore, J.W. and Bourque, P. (2005) *Chapter 2: Software Requirements*, Available at http://www.swebok.org/ch2.html, accessed in May 2008

Al-Rawas, A. and Easterbrook, S. (1996) *Communication problems in requirements engineering: a field study*. Proceedings of the First Westminster Conference on Professional Awareness in Software Engineering, Royal Society, London, 1-2 February 1996.

Arayici, Y., Aouad, G. and Ahmed, V. (2005) *Requirements engineering for innovative integrated ICT systems for the construction industry.* Construction Innovation, 5, 179-200.

Arayici, Y. and Aouad, G. (2005) *Computer integrated construction: an approach to requirements engineering. Engineering,* Construction and Architectural Management, 12 (2), 194-215.

Asanghanwa, E., Davis, A.M., Dieste, O., Hickey, A.M., Juristo, N. and Moreno, A. (2005) *Requirements engineering research: a microcosm of international economic trends.* Available at www.grise.upm.es/miembros/oscar/documents/refsq05.pdf, accessed in May 2008

August, J.H. (1991) *Joint Application Design – The Group Session Approach to System Design*, USA: Prentice Hall

Bahill, A. T. and Henderson, S. J. (2004), *Requirements development, verification and validation exhibited in famous failures,* Systems Engineering, 8 (1):1-14.

Benyon, D. and Skidmore, S. (1987) *Towards a Tool Kit for the Systems Analyst.* The Computer Journal, 30 (1), 2-7.

Boehm, B. (1981) *Software Engineering Economics.* London: Prentice-Hall.

Boonramphai, S. (2008) *Knowledge acquisition of Thai metal forming die designer leader.* Available from: http://www.pantown.com/board.php?id=22315&area=3&name=board3&topic=18&action=view, accessed in June 2008

Borenstein, S. (1991) *Programming As If People Mattered,* New York: Princeton University Press.

Carr, J.J. (2000) *Techniques: requirements engineering and management: the key to designing quality complex systems*, The TQM Magazine, 12 (6), 400-470.

Charvat J. (2003) *Project management methodologies: selecting, implementing, and supporting methodologies and processes for projects, Youth Communicates Series,* John Wiley and Sons.

Chatzoglou, P.D. and Macaulay, L.A. (1996) *Requirements capture and analysis: a survey of current practice.* Requirements Engineering, 1 (2), 75-87.

Christel, M.G. and Kang, K.C. (1992) *Issues in requirements elicitation.* Technical Report, CMU/SEI-92-TR-012. Software Engineering Institute, Carnegie Mellon University.

Chung, L., Nixon, B., Yu, E. & Mylopoulos, J. (2000) *Non-Functional Requirements in Software Engineering*. Boston: Kluwer Academic Publishers.

Coughlan, J., Lycett, M. and Macredie, R.D. (2003) *Communication issues in requirements elicitation: a content analysis of stakeholder experiences,* Information and Software Technology, 45, 525-537.

Coulin, C. and Zowghi, D. (2005) *Requirements Elicitation for Complex Systems: Theory and Practices.* In Requirements Engineering for Sociotechnical Systems, Mate, J.L. and Silva, A. (eds). London: Information Science Publishing.

Dahlstedt, A.G., Karlsson, L., Persson, A., Dag, J.N. and Regnell, B. (2004) *Market-Driven Requirements Engineering Processes for Software Products - a Report on Current Practices.* Available from http://serg.telecom.lth.se/research/publications/publication.php?main!filter!Publications:ID=258, accessed in August 2008

Davis, A. (1992). *Operational Prototyping: A New Development Approach.* Software, 9(5): 70-78.

Davis, A. M. (1993) *Software Requirements: Objects, Functions, & States*; New Jersey, USA: Prentice-Hall, Inc.

Davis, A., and Zowghi, D. (2005) *Good Requirements Practices Are Neither Necessary Nor Sufficient to appear.* Requirements Engineering Journal, 11(1), 1-3.

De Waele, J.P. and Harré, R. (1979) *Autobiography as a psychological method.* In: Ginsberg, G. (Ed.), Emerging Strategies in Social Psychological Research. Wiley, Chichester.

Ding, E.Y. (2005) *Software Models and Analysis: Chapter 11: Requirements Elicitation.* Available at http://software.nju.edu.cn/eryuding/course/RE2004fall/Chapter%2011%20Requirements%20Elicitation.ppt, accessed in August 2008

Dorfman, M. (1999) *Requirements Engineering*, SEI Interactive, 03/99. Available at http://www.sei.cmu.edu/interactive/Features/1999/March/Background/Background.mar99.htm, accessed in August 2008

Easterbrook S. (2004) *What is Requirements Engineering?*, Available at http://www.cs.toronto.edu/~sme/papers/2004/FoRE-chapter01-v7.pdf, accessed in May 2007

Faulk, S. et al., Brackett J., Ward P., and Kirby J. Jr. (1992) *The core method for real-time requirements*, IEEE Software, 9 (5), September.

Finkelstein, A. (1994) *Requirements engineering: a review and research agenda*. Available at http://www.cs.ucl.ac.uk/staff/A.Finkelstein/papers/rereview.pdf, accessed in August 2008

Finn, M., Elliott-White, M. and Walton, M. (2000) *Tourism & Leisure Research Methods: Data Collection, analysis and interpretation*. Harlow: Pearson.

GAO (US General Accounting Office) (1992) *Mission Critical Systems: Defense Attempting to Address Major Software Challenges*, GAO/IMTEC-93-13, December.

Garmer, K., Ylven, J. and Karlsson, I.C.M. (2004) *User participation in requirements elicitation comparing focus group interviews and usability tests for eliciting usability requirements for medical equipment: a case study.* International Journal of Industrial Ergonomics, 33, 85-98.

Goguen, J. and Jirotka, M. (1993) *Techniques for requirements elicitation. Proceedings, Requirements Engineering '93*, edited by Stephen Fickas and Anthony Finkelstein, IEEE Computer Society, pages 152-164.

Goguen, J. and Jirotka, M. (Ed.). (1994) *Requirements Engineering: Social and Technical Issues*. London: Academic Press.

Goguen, J. and Linde, C. (1993) *Techniques for Requirements Elicitation.* 1st IEEE International Symposium on Requirements Engineering (RE'93), San Diego, USA, 4-6th January 1993, pp. 152-164.

Gray, H. and Campbell, J. (2002) *The Electronic Enablement of Businesses in Regional Thailand.* Available at http://203.159.5.16/digital_gms/Proceedings/C65_HEATHER_GRAY.pdf, accessed in July 2008

Grunbacher, P. and Briggs, B. (2001), *"Surfacing tacit knowledge in requirements negotiation: experiences using EasyWinWin",* Proceedings of the Hawaii International Conference on System Sciences, IEEE Computer Society, Los Alimotos, CA.

Harwell, R., Aslaksen, E., Hooks, I., Mengot, R. and Ptack, K. (1993*) What is a requirement?*, Proc. 3rd Ann. Intl. Symp. Natl. Council Systems Eng.

Hickey, A.M. and Davis, A.M. (2002) *Requirements Elicitation and Elicitation Technique Selection: A Model for Two Knowledge-Intensive Software Development Processes*, Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03).

Hickey, A.M., Davis, A.M. and Kaiser, D. (2003) *Requirements Elicitation Techniques: Analyzing the Gap between Technology Availability and Technology Use*, Comparative Technology Transfer and Society, 1 (3), 279-302.

Hall, T., Beecham, S. and Rainer, A. (2002) *Requirements problems in twelve software companies: an empirical analysis. IEE Proc.-Softw*, 149 (5), 153-160.

Hofstede G. (1991) *Cultures and Organizations: Software of the Mind.*, UK: McGraw-Hill.

Holmes, H. and Tangtongtavy, S. (1995) *Working with the Thais*, White Lotus, Thailand.

Hsia, P., Davis, A. and Kung, D.C. (1993) *Status report: requirements engineering,* IEEE Software, 10 (6), 75–79.

Humphrey, W.S. (1989) *Managing the Software Process*. Reading, Massachusetts: Addison-Wesley Publishing Company.

INCOSE (2000) *How To: Guide for all Engineers*, Version 2, International Council on Systems Engineering.

Jackson, M. (2001) *Problems, methods and specialisation*. Special Issue of SE Journal on Software Engineering in the Year 2001.

Japan Bank for International Cooperation (2004), Release of Ex-Post Evaluation Report for ODA Loan Projects 2003 (Summary) Toward Increased Effectiveness and Credibility of ODA Loans. Available from: http://www.jica.go.jp/english/news/jbic_archive/autocontents/english/news/2004/000045/index.html, accessed in September 2008

Jennings, G. (2001) *Tourism research*. New York: John Wiley and Son Inc.

Jirachiefpattana, W. (1996) *The Impact of Thai Culture on Executive Information Systems Development*, Proceedings of the 6th International Conference Theme 1, Globalization: Impact on and Coping Strategies in Thai Society, 14-17 October, Chiang Mai, Thailand, pp 97-110.

Jones, C. (1991) *Applied Software Measurement*, Second Edition, New York: McGraw Hill.

Johns Hopkins University (JHU) (1998) *Part-Time Programs in Engineering and Applied Science*, Whiting School of Engineering catalog, JHU, Baltimore, MD.

Karlsson, L., Dahlstedt, A.G., Regnell, B., Dag, J.N. and Persson, A. (2007) *Requirements engineering challenges in market-driven software development – an interview study with practitioners*. Information and Software Technology, 49, 588-604.

Karten, N. (1994), *Managing Expectations*, New York, NY: Dorset House Publishing.

Kauppinen, M., Vartiainen, M., Kontio, J., Kujala, S. and Sulonen, R. (2004) *Implementing requirements engineering processes throughout organizations: success factors and challenges.* Information and Software Technology, 46, 937-953.

Kelly, G. (1955). *The psychology of personal constructs*. New York: Norton.

Keretho, S. (2008) *Software Development Life Cycle, from acquisition to development, operations and maintenance*. Available from: http://www.07.ibm.com/th/events/qm/download/SW_Development_Life_Cycle_Dr_Somnuk_Keretho.pdf, accessed in November 2008

Kjaer, A. and Madsen, K. (1995) *Participatory analysis of flexibility,* Communications of the ACM, 38 (5), 53–60.

Korpela, M., Soriyan, H.A., Olufokunbi, K.C. and Mursu, A. (1998) *Blueprint of an African systems development methodology: an action research project in the health sector,* in Avgerou, C. (eds) (1998) Proceedings of the Fifth International Working Conference of IFIP WG 9.4 – Implementation and Evaluation of Information Systems in Developing Countries, February 18-20, pp. 273-285.

Kotler P., Kartajaya H., and Huan H. D. (2007) *Think ASEAN! Rethinking Marketing toward ASEAN Community 2015*, Singapore: McGraw-Hill, pp. 192.

Kotonya, G. and Sommerville, I. (1998) *Requirements Engineering: Processes and Techniques*. New York: Viley

Kumarnbun, P. (2007) *Managing Innovation*, Available from: http://www.hitsplay.com/blog/DisplayStoryDetail.aspx?blogID=5187&storyID=1370 accessed in November 2008

Lamsweerde, A. van (Axel), (2009) *Requirements engineering : from system goals to UML models to software specifications / Axel van Lamsweerde,* Chichester, England ; Hoboken, NJ : John Wiley,

Laudon, K.C. and Laudon, J.P. (1995) *Essentials of Management Information Systems.* New York, MacMillan Publishing Co.

Loucopoulos, P. and Karakostas, V. (1995) *System Requirements Engineering*, Berkshire, UK: McGraw-Hill Book Company Europe.

Macaulay, L.F., Chris, K.M. and Hutt, A. (1990) *USTM: A New Approach to Requirements Specification*, Interacting with Computers,. 2 (1), 92-118.

Maiden, N., Minocha, S., Sutcliffe, A., Manuel, D., and Ryan, M. (1999) *A co-operative scenario based approach to acquisition and validation of system requirements: How exceptions can help!,* Interacting with Computers, 11 (6), 645-664.

Maiden, N. (1998). *CREWS-SAVRE: Scenarios for Acquiring and Validating Requirements.* Automated Software Engineering, 5 (4), 419-446.

Maiden, N. and Rugg, G. (1996*), ACRE: Selecting Methods For Requirements Acquisition.* Software Engineering Journal, 11 (3), 183-192.

Martin, J. (1984) *Information System Manifesto*. Englewood Cliffs, NJ: Prentice Hall.

Martin, J. (1988) *Information Engineering, Book I: Introduction*. Englewood Cliffs, NJ: Prentice Hall.

Marasco. J. (2006) *The Software Development Edge: Essays on Managing Successful Projects.* Addison-Wesley

McGraw, K.L. and Harbison-Briggs, K. (1989) *Knowledge acquisition: Principles and guidelines*. New Jersey: Prentice-Hall.

Mead, N.R. (2003) *Requirements Engineering for Survivable Systems*, Technical Note CMU/SEI-2003-TN-013. Available at http://www.sei.cmu.edu/publications/pubweb.html, accessed in November 2008

Mochal, T. (2006) *Define your project boundaries in the up-front scope definition*, Available at
http://articles.techrepublic.com.com/5100-10878_11-6059454.html, accessed in January 2009

Mongkolpong P. (2008) *Failure of product value misunderstanding*, Available from: http://www.thannews.th.com/detialNews.php?id=M2823022&issue=2302, accessed in June 2008

Mu, K., Jin, Z., Lu, R., Peng, Y. (2006) *Handling non-canonical software requirements based on Annotated Predicate Calculus.* Knowl Inf Syst 11 (1), 85–104.

Naoum, S.G., (1998) *Dissertation Research and Writing for construction students*. Oxford: Butterworth-Heinemann

Nasr, E. (2005) *Challenges in Requirements Engineering for Embedded Systems.* In Requirements Engineering for Sociotechnical Systems, José Luis Maté (Ed), Melbourne: Information Science Publishing.

NATURE team (1996) *Defining visions in context: models, processes and tools for requirements engineering,* Information System, 21 (6), 515-547.

Nuseibeh, B. and Easterbrook, S. (2000) *Requirements engineering: A roadmap.* Proceedings of the conference on The Future of Software Engineering, Limerick, Ireland.

Nuseibeh, B. and Easterbrook, S. (2007) *Requirements engineering: a roadmap*, Available at http://mcs.open.ac.uk/ban25/papers/sotar.re.pdf, accessed in January 2009

Oppenheim, A.N., (1992) *Questionnaire Design Interviewing and Attitude Measurement.* New York: Continuum.

Orlikowski, W.J. (1993) *CASE Tools as Organizational Change: Investigating Incremental & Radical Changes in Systems Development*. MIS Quarterly, 17 (3), 309 - 340.

O'Hara, F. (2000) *European experiences with software process improvement*, Proceedings of the 22nd International Conference on Software Engineering, (ICSE 2000), 2000, pp. 635–640.

Paniangvait P. (2007) Tom *Yum Goong and Pork Flovour : The popular product of Mama*, Available at: http://www.brandage.com/Modules/DesktopModules/Article/ArticleDetail.aspx?tabID=2&ArticleID=1398&ModuleID=21&GroupID=520, accessed in January 2009

Phasukvanich, P. (2003) *Money Squandered by 'Poor Preparation' (Summary) Board Warns of Huge Social, Economic Costs*, Available at: http://www.burmalibrary.org/docs/SW14.htm, accessed in March 2007

Pongpaibul, M. and Boehm, B. (2005) *Improve quality through software process improvement in Thailand: initial analysis*. Available at: http://portal.acm.org/citation.cfm?id=1082983.1083299., accessed in January 2007

Przybilski, M. (2006) *Requirements Elicitation in International Research Projects.* Proceedings of the Twelfth Americas Conference on Information Systems, Acapulco, Mexico August 4-6th 2006.

Rajagopal, P., Lee, R, Ahlswede, T., Chia-Chu Chiang and Karolak, D. (2005). *A new approach for software requirements elicitation.* Available from http://ieeexplore.ieee.org/Xplore/login.jsp?url=/ielx5/9806/30915/01434864.pdf?arnumber=1434864, accessed in February 2007

Ralph R. Young.,(2004), *The requirements engineering handbook*, Boston, MA. Artech House

Robertson, S. and Robertson, J. (1999) *Mastering the requirements process*. Great Britain: Addison-Wesley.

Rohitratana, K. (1998) *The role of Thai values in managing information systems: a case study of implementing an MRP systems*, in Avgerou, C. (eds) (1998) Proceeding of the Fifth International Working Conference of IFIP WG 9.4 – Implementation and Evaluation of Information Systems in Developing Countries, February 18-20. London School of Economics and Political Science and Asian Institue of Technology Bangkok, IFIP, pp.188-201.

Rossi, M. and Tuunanen, T. (2004) *A Method and Tool for Wide Audience Requirements Elicitation and Rapid Prototyping for Mobile Systems*. ER Workshops 2004, LNCS 3289, pp.629-640.

Rush, G. (1985) *A fast way to define system requirements*. Computerworld, 7th October, 11-16.

Ryan Ian R Faulconbridge and Michael J. Ryan (2002) *Managing complex technical projects: a systems engineering approach*, Artech House

Sage, A.P. and Palmer, J.D. (1990) *Software Systems Engineering*. New York: John Wiley & Sons.

Saiedian, H. and Dale, R. (2000) Requirements engineering: making the connection between the software developer and customer. *Information and Software Technology*, 42, 419-428.

Santander, V.F.A. and Silva, D.R. (2007) *Requirements Engineering Contributions on the Development of Educational Software for the Blind or People with Impaired Vision – An Experience Account,* Available at www.clei.cl/cleiej/papers/v9i1p6.pdf, accessed in February 2007

Sawyer, P. and Kotonya, G. (2001) *Software Requirements," 9-34. Guide to the Software Engineering Body of Knowledge, Trial Version 1.00 (SWEBOK)*. Los Alamitos, CA: IEEE Computer Society, <http://www.swebok.org/>.

Shaw, M. and Gaines, B. (1996) *Requirements Acquisition.* Software Engineering Journal, 11 (3), 149-165.

Sommerville, I. and Sawyer, P. (1997) *Requirements Engineering: A Good Practice Guide*, England: John Wiley & Sons.

Saunders, M., Lewis, P., .Thornhill, A. (2003) *Research methods for business students.* 3rd ed., Gosport, Ashford Colour Press Ltd.

Sorod, B. (1991) *The Influence of national and Organizational Cultures on Managerial Values, Attitudes, and Performance (National Cultures)*, The University of Utah, 1991

Standish Group (1995) *The Chaos Report*. Available at www.standishgroup.com, accessed in April 2007

Stevens, R., Brook, P., Jackson, K. and Arnold, S. (1998) *Systems Engineering: Coping with Complexity*. Prentice Hall Europe.

Suzanne R. (2000) *Project Sociology: Identifying and involving the stakeholders,* The Atlantic Systems Guild Ltd.

Thanasankit T. (2002) *Requirements engineering - exploring the influence of power and Thai values*. European Journal of Information Systems, 11 (2), 128-141.

Thanasankit, T. and Corbitt, B. (2000) *Cultural context and its impact on requirements elicitation in Thailand,* EJISDC, 1 (2), 1-19.

Thayer, R.H. and Dorfman, M. (1997) *Software Requirements Engineering*, IEEE Computer Society Press, 176-205, 389-404.

Tuunanen, T. (2003) A new perspective on requirements elicitation methods. *Journal of Information Technology Theory and Application*, Available at http://findarticles.com/p/articles/mi_qa4008/is_200301/ai_n9241764/pg_7, accessed in June 2007

Van Lamsweerde, A., Darimont, R. & Letier, E. (1998) *Managing conflicts in goal-driven requirements engineering*. IEEE Transactions on Software Engineering, 24 (11), 908-926.

Van Lamsweerde, A. and Letier, E. (2000) *Handling obstacles in goal-oriented requirements engineering.* IEEE Transactions on Software Engineering, Oct 2000, 26 (10), 978-1005.

Veal, A. J. (2003) *Research methods for leisure and tourism.* 3rd ed. Essex, Financial Times Prentice Hall.

Viller, S. and Sommerville, I. (1999) *Social Analysis in the Requirements Engineering Process: from ethnography to method.* 4th International Symposium on Requirements Engineering (RE'99), Limerick, Ireland, 7-11th June 1999.

White, B., (2000) *Dissertation Skills for Business and Management Students*. London: Thomson.

Whitten, N. (2005) *Neal Whitten's No-Nonsense Advice for Successful Projects*. Vienna, VA: Management Concepts.

Wiegers, K. Eugene, (1999) *Software requirements,* Redmond, Wash. Microsoft Press

Yongpisarnpob A. (2007) *Through the heart of 7-11 mini-mart,* Available from: http://www.gotomanager.com/news/details.aspx?id=1348, accessed in February 2009

Young, R.R. (2002) *Recommended Requirements Gathering Practices.* Available from http://www.stsc.hill.af.mil/crosstalk/2002/04/young.html., accessed in August 2008

Zave, P. (1997) *Classification of research efforts in requirements engineering.* ACM Computing Surveys, 29(4), 315-321.

Zikmund, W. G. (1997) *Business research methods*. Orlando, The dryden press.

SUPPORTING READINGS

Gause, D.C. and Weinberg, G.M. (1989) *Exploring Requirements: Quality before Design*. New York, NY: Dorset House.

Kaewkasi, C. (2002) *Conceptualization of aspect-oriented requirements model*, Master's Degree Thesis, Chulalongkorn University Thesis, Thailand.

Kautz, K., Hansen, H. and Thaysen, K. (2000) *Applying and adjusting a software process improvement model in practice: the use of the IDEAL model in a small software enterprise*, Proceedings of the 22nd International Conference on Software Engineering, (ICSE 2000), 2000, pp. 626–633.

Potts, C. (1991), *Seven (plus or minus two) Challenges for Requirements Research*, Sixth International Workshop on Software Specification and Design, IEEE Computer Society Press, Los Alamos CA, 1991, pp. 256–259.

Vatanawood, W. (2002) *Formal specification synthesis using requirements particle networks*. Master Degree Thesis, Chulalongkorn University, Thailand.

Zahran, S. (1998) *Software Process Improvement: Practical Guidelines for Business Success*. England, Harlow: Addison-Wesley.

Appendix A: The Questionnaire

Dear Sir/Madam,

As a Master's Student at University of Technology Sydney (Australia), I am carrying out a research under the topic of **'INVESTIGATING THE STATE OF REQUIREMENTS ELICITATION IN THAILAND'.** Please complete this questionnaire as accurately as possible so that I can draw some realistic conclusions. All information is strictly for academic purposes and it will remain confidential. Thank you for your assistance. The followings are description and concept of requirement engineering that might help you to understand more about requirement engineering and facilitate the completion of the questionnaire.

Requirement Engineering

Requirements engineering (RE) is treated either as: (i) a branch of systems engineering whose ultimate goal is to deliver some systems behavior to its stakeholders; (ii) a branch of software engineering concerned with the real-world goals for functions of and constraints on software system.

In the context of software engineering, requirement engineering is the discipline of determining, analyzing, pruning, documenting, and validating the needs and requirements of stakeholders for a software system. It proposes a set of techniques, methodology, and tools capable of giving technical support and help on the production of quality software.

In the context of system engineering requirements engineering focuses on improvements to the front-end of the system development life-cycle by establishing the needs that have given rise to the development process and organizing this information in a form that will support system conception and implementation. It is a phase in system development that entails all the activities of user requirements identification, analysis of the so collected requirements to identify more requirements, converting the requirements into systems requirements specification document (SRS document) and then validating and verifying the SRS document in relation to the user needs.

Requirements engineering is described using a four-step process model as shown in the Figure, including requirements elicitation, analysis and negotiation, documentation, and validation.

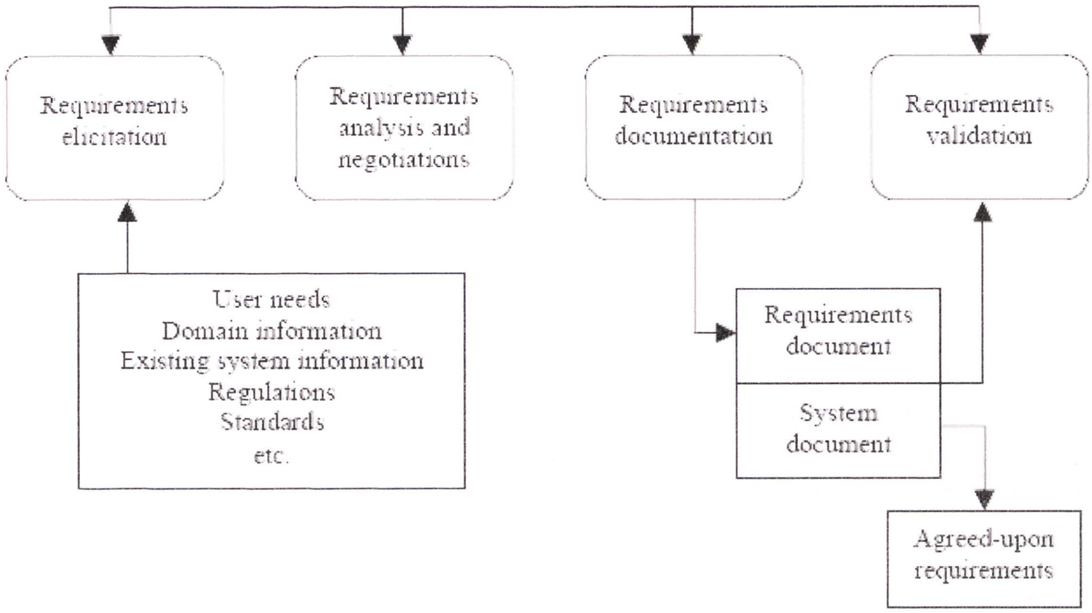

Figure A1: Four-step process of requirement engineering

Requirements elicitation is reported as the first stage in building an understanding of the problem that the software is required to solve. Requirements analysis has to do with the process of analyzing requirements to detect and resolve conflicts among requirements, discover the bounds of the system and how it must interact with its environment, and elaborate user requirements to software requirements. Requirements negotiation involves with resolving conflicts, i.e. those that might occur between stakeholders, or between requirements and resources. Requirements validation is involved with checks for omission, conflicting requirements, and ambiguities.

<u>Section 1</u>: Business information

1. What kind of main business does your organization operate?

☐ Software development ☐ Construction

2. Is your organization a national or international one?

☐ National organization (operate only in Thailand)

☐ International organization

3. What is the size of your organization?

☐ Less than 50 employees ☐ 50-200 employees

☐ More than 200 employees

4. What is the value of your organization's investment capital?

☐ Less than 50 million Baht ☐ 50-200 million Baht

☐ More than 200 million Baht

5. How long has your organization been established?

☐ Less than 1 year ☐ Between 1 to 3 years

☐ Between 3 to 6 years ☐ More than 6 years

<u>Section 2</u>: Demographic information

6. Occupation

☐ Software engineer ☐ System engineer

☐ Manager ☐ Requirements engineer

☐ Supervisor ☐ Other ____________

7. How long have you been software engineer and/or been involved in requirement engineer at your current organization?

☐ Less than one year ☐ Between 1 to 3 years

☐ Between 3 to 6 years ☐ More than 6 years

8. What is your main role and responsibility in your organization?

____

____

____

____

____

<u>Section 3</u>: Requirements elicitation

9. Have you ever worked in the field of requirements engineering or been involved with requirements elicitation?

☐ Yes, always ☐ Yes, sometimes

☐ Not sure (cannot remember) ☐ Never

10. How do you rate the performance of the requirements elicitation process at your organization?

☐ Very well performed ☐ Well performed

☐ Unsure ☐ Poorly performed

☐ Very bad

11. Please rank sources of requirements knowledge that you acquire for system development? (From "1" which is the most important source to "5" which is the least important)

☐ Customers or Users' needs

☐ Domain information

☐ Existing system information

☐ Other stakeholders' needs

☐ Others..................................

12. What is the most frequently used technique for requirements elicitation in your organization?

☐ Prototyping

☐ Traditional techniques such as interview, questionnaire survey, analysis of existing documentation

☐ Group elicitation techniques such as brainstorming, focus groups, workshops, and organizational game

☐ Model-driven techniques such as goal-based methods, scenario-based methods

☐ Cognitive techniques such as protocol analysis, laddering, card sorting, repertory grids

☐ Contextual techniques such as ethnographic techniques (i.e. participant observation), ethno methodology, conversation analysis

☐ Others..................................

13. Have you ever had any problem or difficulty in the process of requirements elicitation?

☐ Yes, please specify ....

☐ No

Section 4: States of requirements elicitation

Below is a list of statements. Please indicate your opinions by ticking the appropriate box below.

Statements	Strongly agree	Agree	Not sure	Disagree	Strongly disagree
End users/customers requirements					
- You regularly discuss with the end users/customers about their requirements.	☐	☐	☐	☐	☐
- You completely obtain all the necessary requirements information from the end users/customers.	☐	☐	☐	☐	☐
- The end users/customers requirements are always changed before the system development is finished.	☐	☐	☐	☐	☐
- When you come to the process of requirements elicitation from end users/customers, you are concerned with what they do not want as well as what they want.	☐	☐	☐	☐	☐
- It can be said that in the process of requirements elicitation, end users/customers requirements are the most important	☐	☐	☐	☐	☐

Statements	Strongly agree	Agree	Not sure	Disagree	Strongly disagree
Stakeholders requirements					
- Apart from end-users/customers, you have a chance to discuss with other stakeholders about their requirements.	☐	☐	☐	☐	☐
- You regularly discuss with other stakeholders about their requirements.	☐	☐	☐	☐	☐
- You take other stakeholders requirements into consideration for the system development as well as end users/customers' requirements.	☐	☐	☐	☐	☐
Domain-imposed (functional) requirements					
- You completely obtained all domain-imposed (functional) requirements for any system you have developed	☐	☐	☐	☐	☐
- Techniques, technology and software used for eliciting domain-imposed (functional) requirements are updated	☐	☐	☐	☐	☐
- The complexity of software system complicates the requirements elicitation process	☐	☐	☐	☐	☐

Statements	Strongly agree	Agree	Not sure	Disagree	Strongly disagree
Management support/commitment					
- Management provide strong leadership and support for requirement elicitations	☐	☐	☐	☐	☐
- Management establishes requirements elicitation practices as an integral part of the system development process	☐	☐	☐	☐	☐
- Requirement elicitation processes and practices are standardized across the projects done by the company	☐	☐	☐	☐	☐
- Management at all levels of the organization supports the requirement elicitation	☐	☐	☐	☐	☐
- Management is willing to participate in meetings with end-users/stakeholders	☐	☐	☐	☐	☐
- Management is committed to provide training and resources for requirements elicitation implementation	☐	☐	☐	☐	☐

Statements	Strongly agree	Agree	Not sure	Disagree	Strongly disagree
Communication					
- You always communicate goals, objectives, tasks, procedures, constraints, timetables to end-users/stakeholders regularly	☐	☐	☐	☐	☐
- You are always listening to what customers need in order to impress them	☐	☐	☐	☐	☐
- You always use verbal communication with end-users/stakeholders	☐	☐	☐	☐	☐
- You always use written communication with end-users/stakeholders	☐	☐	☐	☐	☐
- In the process of requirements elicitation, there are intermediates between you and the end users/customers.	☐	☐	☐	☐	☐
Requirements documentation					
- Requirement documents are laid out in a highly readable manner.	☐	☐	☐	☐	☐
- All of the necessary information is provided in requirements documents, even if only by reference	☐	☐	☐	☐	☐
- Requirements documents clearly explain why the system is required	☐	☐	☐	☐	☐
- Glossary is always included in the requirements documents	☐	☐	☐	☐	☐

- All of the services provided and functions performed by the system are discussed in the requirements documents	☐	☐	☐	☐	☐
- All constraints in the design are discussed in the requirements documents	☐	☐	☐	☐	☐

Section 5: **Problems of requirements elicitation**

5.1 Frequency of problems

Below is a list of problems. Please indicate your opinions by giving the score of frequency at which the following problems occur in your organization.

Problems	Very rare	Sometimes	Moderate frequently	Often	Very often
- Requirements do not reflect the actual needs of the users/system stakeholders	☐	☐	☐	☐	☐
- Inconsistent requirements (i.e. requirements are stated in different and inconsistent ways in different parts of the document)	☐	☐	☐	☐	☐
- Incomplete requirements	☐	☐	☐	☐	☐
- Conflicting requirements (i.e. requirements come from different sources and they are not compatible to each other)	☐	☐	☐	☐	☐
- Misunderstood or misinterpreted requirements (i.e. different stakeholder communities use different jargon or different definitions for the same words).	☐	☐	☐	☐	☐
- Ambiguous (i.e. words are ambiguous)	☐	☐	☐	☐	☐
- Vague requirements (i.e. statements of the requirements fail to elicit an image or understanding of what is desired in the reader)	☐	☐	☐	☐	☐