A STUDY OF REQUIREMENTS ENGINEERING PRACTICES AMONG SOFTWARE DEVELOPERS AT UUM INFORMATION TECHNOLOGY (UUMIT)

INAM ABDULLAH LAFTA

MASTER OF SCIENCE (INFORMATION TECHNOLOGY)
UNIVERSITI UTARA MALAYSIA
2015

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences

UUM College of Arts and Sciences

Universiti Utara Malaysia

06010 UUM Sintok

Abstrak

Kejuruteraan Keperluan Perisian (RE) adalah satu proses yang sistematik dan bersepadu untuk mendapatkan, menjelaskan, merundingkan, menentu sahkan dan menguruskan keperluan sistem dalam projek pembangunan perisian. Pengurusan UUM dibantu oleh pelbagai sistem dalam akademik, pentadbiran, urusan pelajar dan lain-lain lagi. Kebanyakan sistem ini dibangunkan dan diselenggarakan oleh Jabatan Teknologi Maklumat (UUMIT). Tujuan kajian ini adalah untuk mengkaji semula amalan kejuruteraan keperluan perisian semasa dan mencadangkan amalan keperluan kejuruteraan yang sepatutnya diamalkan ketika membangunkan perisian di UUMIT. Di samping itu, penggunaan khidmat luar bagi pembangunan perisian makin berkembang kerana manfaatnya yang besar dalam mengatasi masalah sumber-sumber yang terhad di organisasi. Masalah utama yang dibincangkan dalam kajian ini adalah kekurangan kajian yang menyokong aktiviti pembangunan perisian di Jabatan Teknologi Maklumat (UUMIT). Kajian ini menggunakan kaedah kuantitatif dan kajian literatur yang sistematik untuk menjawab persoalan kajian. Kepentingan utama kajian ini adalah ia dapat membantu institusi pendidikan untuk menghasilkan pembangunan perisian yang berkualiti serta menjimatkan kos dan masa dengan melaksanakan amalan kejuruteraan keperluan perisian yang baik. Selain itu, kajian ini memberi sumbangan kepada UUM dengan mengenal pasti aktiviti yang perlu dijalankan untuk pembangunan perisian supaya pihak pengurusan dapat memperuntukkan bajet bagi menyediakan latihan yang mencukupi dan tepat serta seminar untuk pembangun perisian. Penyelidik telah mengkaji tiga pemboleh ubah; *Requirements Description, Requirements Development (Requirements Elicitation, Requirements Analysis and Negotiation, Requirements Validation), and Requirement Management*. Hasil daripada kajian menunjukkan bahawa amalan semasa kejuruteraan keperluan perisian di jabatan UUM IT adalah menggalakkan, tetapi perlu dipertingkatkan kerana kebanyakan amalan RE yang berkaitan dengan perkembangan keperluan dan pengurusan keperluan dijalankan secara biasa dan tidak kerap. Penyelidik mengesyorkan supaya program latihan yang efektif disediakan untuk kakitangan UUMIT tentang amalan RE dan meningkatkan pemahaman mereka mengenai keperluan sistem menggunakan amalan RE untuk membangunkan sistem yang lebih baik untuk universiti. Kajian lanjut diperlukan pada masa akan datang untuk memahami kesan amalan RE lain dalam pembangunan perisian.

Keyword: Kejuruteraan Keperluan Perisian, merundingkan, UUM IT, metodologi kuantitatif, kajian literatur yang sistematik

Abstract

Requirements Engineering (RE) is a systemic and integrated process of eliciting, elaborating, negotiating, validating and managing the requirements of a system in software development project. UUM has been supported by various systems in academic, administrative, students' affair and many others. Most of the current systems are developed and maintained by the Information Technology Department (UUMIT). The aim of this study is to review the current requirements engineering practices and proposing requirements engineering practices during software development at UUMIT. The outsourcing of software development is rapidly growing because of its allied benefits in the limited resources of the organizations. The main problem that is discussed in this research is the lack of studies that support software development activities at the Information Technology department (UUMIT). The study used quantitative methodology and systematic literature review to answer research questions. The main significance of this study is helping educational institutes to produce quality software development and saving cost and time by implementing requirements engineering practices. In addition to that, the study contributes to UUM by identifying the activity needed for software development so that the management is able to allocate budget to provide adequate and precise training as well as seminars for the software developers. The researcher investigated three variables; Requirements Description, Requirements Development (Requirements Elicitation, Requirements Analysis and Negotiation, Requirements Validation), and Requirement Management. The results from the survey showed that the current practice of requirement engineering in IT department of UUM is encouraging, but need for further development because most of RE practices associated with requirement development and requirement management are achieved on a regular basis and not frequently. The researcher recommended providing effective training programs for UUMIT staffs on RE practices and increases their understanding on system requirements using RE practices to develop better systems for the university. Further investigation is required in the future to understand the effect of other RE practices on software development.

Keywords: Requirements engineering practices, negotiating, UUMIT, quantitative methodology, systematic literature review

Acknowledgement

Thanks to Allah

To the Memory of My Father and Brother...

To My Mother...

To My Family and Friends...

I am grateful to ALLAH for the good health and wellbeing that were necessary to complete this research. I place on record, my sincere thank you to Dr. Azhem hussain, for sharing his expertise, and sincere and valuable guidance and encouragement extended to me. I wish to express my sincere thanks to UUM and all the Faculty member in school of computing especially Committee members, Session Chair Dr. Mohd Hasbullah Omar, AP Dr. Haslina Mohd and Dr. Hazaruddin Harun for providing me guidance with all the necessary facilities for the research. I also thank my family, my colleagues and friends for the encouragement unceasing, support and attention.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

RE	Requirements Engineering
RD	Requirements Development
RM	Requirements Management
UUM	Universiti Utara Malaysia
UUMIT	Universiti Utara Malaysia Information Technology
SDLC	Software Development Life Cycle
SREM	Software Requirement Engineering Method
SLR	Systematic Literature Review

CHAPTER ONE
INTRODUCTION

1.1 Background

Nowadays, software applications have considerably supported our work and daily life (Insfran, Chastek, Donohoe, & do Prado Leite, 2013) . Software applications are everywhere, at the work place, in the office, at home, in the car and many other places. Moreover, almost all electrical equipments contain software applications leading to the increase in usage of software in electrical appliances significantly. Today, software is used across various industries, as in education, agriculture, health, financial, economics, entertainment and others.

Certainly, there is an urgent need to develop a standard software that may satisfy the needs of the users without any error (s, Srinivasan, Dravid, kasera, & Sharma, 2014). Requirement of software is fulfilled by the requirements engineering (RE) which is the process of determining requirements (Cheng & Atlee, 2009) Moreover, Cheng and Atlee (2009) mentioned that successful Requirements Engineering (RE) involves understanding of the stakeholders needs; considerate software contexts; modeling, analyzing, negotiating, as well as supporting stakeholders' requirements; assessing documented requirements; and managing the requirements. There are many researches that identify the need of development of a quality software that may meet needs and objectives of the customers and give value to a stakeholder (Khan, Naz'ri bin Mahrin, & bt Chuprat, 2014; K. Wiegers, 2013).

Main focus of requirement engineering is on requirements and needs to reach better final products (Sheikh, Dar, & Sheikh, 2014). Asghar and Umar (2010) pointed out that RE is acknowledged as the first phase of software engineering process and it is considered as one of the main phase in software development. Moreover, Khan *et al.* (2014) and Shah and V Patel (2014) asserted that, unclear requirement engineering is the main reason of software project failure. Where, Khan et al. (2014) said that "requirement engineering phase is difficult and crucial" p.64. Young (2004) stated that, it contributes to product faults. It is affirmed that effective RE is essential during software development. In addition, Mendez Fernandez, Lochmann, Penzenstadler, and Wagner (2011) mentioned that Requirement Engineering is the main factor that affects the productivity as well as product quality.

Thus, it can be stated that RE is an essential phase for software development (Sankhwar, Singh, & Pandey, 2014), and therefore RE practices should be taken into consideration for every software development project. Requirement engineering (RE) is a procedure through which system requirements are determined. RE involves the activities of discovering the needs of stakeholders, understanding the context of requirements, modelling, negotiating, validating, documenting and managing these requirements (Shah & V Patel, 2014). Where, in the process of software development requirement engineering is assumed to be the most important phase (Basharat, Fatima, Nisa, Hashim, & Khanum, 2013).

In this study, the RE is defined based on (Wiegers & Beatty, 2013; Wiegers, 2003) that mentioned RE is composed of two main activities which are requirements development and requirements management. Thus, this study only focuses on how requirements are being elicited, analyzed, negotiated and validated in requirements development. Where, Shah and V Patel (2014) and Insfran et al. (2013) referred that RE is a systemic and integrated process of eliciting, elaborating, negotiating, validating and managing the requirements of a system. According to Kavitha and Thomas (2011), eliciting proper comprehension and management of requirements of a software are main determinants of success in the process of development of software. In addition, from the requirements management activity this study focuses on how to handle changes in the requirements, version control of requirements, traceability and tracking of requirements. Therefore, this study will not consider other aspect or area in RE.

Universiti Utara Malaysia (UUM) is one of the public universities located at the northern part of Malaysia. As an eminent management university, UUM has been supported by various systems in academic, administration, students' affair and many others. Most of the system are developed and maintained by Information Technology UUMIT.In 2015, the Universiti Utara Malaysia replace the name of the Computer Center into UUM Information Technology, therefore in the throughout this study the researcher will used UUMIT. The UUMIT is a center that acts as a heart of the UUM that deliver, support and maintains all the systems. Thus, it is important for UUMIT to deliver quality software in time within the budget. Requirements engineering can support organizations in developing software systems of standard quality in time and within given budget limit and that must reflect true needs of the customers (Nilofer & Sheetal, 2012). UUMIT has its own software developers that develop software in house to support the business function of UUM. Understanding the importance of UUMIT this study attempts to investigate how the software developers at UUMIT practice the RE during software development. As a result, this study is able to highlight how the RE activities have been implemented and how can we help the software developers to improve the RE practices in the future.

1.2 Problem Statement

Main focus of requirement engineering is on requirements and needs to reach better final products (Sheikh et al., 2014). In software development, a project is considered successful whenever it is able to deliver within the time frame considering the budget and has all the specified features as well as functions (The Standish Group International, 2009). In addition, The Standish Group International (2009) reported that there are two main reasons which lead to software project failure. The first reason is requirements and the second reason is poor or incomplete requirements specification (Prior & Keenan, 2005).

Poor specification means ambiguous and requirements are specified generally. Thus, it leads the developer in making incorrect business logic decision. These circumstances should be avoided as it indicates bad business and loss of precious project time.

Furthermore, The Standish Group International, (2009) reported that most of software projects failed because of user input, inadequate and changing requirements and J. Y.-C. Liu, Chen, Chen, and Sheu (2011) also agreed on this. Infact, a software project is successful because of the involvement from user, getting strong support from the management and having clear requirement statements. Therefore, we may define requirement as a significant feature in a system, a statement that can identify a characteristic or quality factor in a system in for having utility and value to the end user. Moreover, Sheikh et al. (2014) and Bashir and Qureshi (2012) asserted good requirement engineering as one of the crucial factors in the success of the projects.

Primarily, the Chaos report has pointed out the factors that contributed to software project failure and those factors are; getting very weak involvement from users as well as from management, having bad project management, ambiguous requirements, scope, and software methodology as well as project estimation (Engineering, of, & Society, 2004). Problems that lead to requirements engineering are common and definitely most significant (Sommerville & Ransom, 2005).

Requirement phase is one of the major phases for the success of a software development life cycle (SDLC). Most of the software has not been completed just because of negligence of requirement phase. Due to increasing vulnerabilities of a system it is one of the challenging tasks for the software (Bokhari & Alam, 2013).

Basharat et al. (2013) acknowledged requirement engineering as a vital phase in removing tardy and unproductive stages in organizations. Shubhamangala, Rao, Dakshinamurthy, and Singh (2012) asserted that software project failures are due to poor Requirement Engineering (RE), and this agreed with a survey conducted by Cerpa and Verner (2009) which investigated that why software projects fail. Their study has shown that the commonly reason for software failure was inadequate requirements that contribute to 73%. Another reason for software failure is unrealistic expectation when the client did not spend enough time with developers in defining the requirements completely. Moreover, the initial requirements are poorly defined and this will result the scope changes during software development. Although the software world has changed significantly with several programming and development paradigms, poor requirements is yet the main reason for software projects failure in this decade, which is also one of the main reasons for software failure in previous decades. Lindquist (2005) reported that 71% of software projects that fail due to poor requirements make it the single biggest reason for project failure. In order to promote software project success, RE plays an active vigorous role in the system development project.

At the present time, many methods and techniques in RE are accessible abundantly for practitioners (Ramingwong, 2012). Yet, several methodologies which are proven successful in a software development may not be appropriate for another. Hence, it is a challenge in selection of methods and techniques as it lead to software failure.

In the previous two decades, problems in requirements engineering are factors that lead to ineffectiveness and failure of most software projects (Sommerville & Ransom, 2005). Most the projects were failed because of poor identification of requirements (El Emam & Birk, 2000). Kumar and Kumar (2011) stated that poor requirements leads to increase the overall cost, decrease quality of the system or fail altogether.

The outsourcing of software development is rapidly growing day by day because of its allied benefits in limited resources of the organizations. However, it is observed that many outsources projects can not achieve anticipated and expected results. The reason behind this failure are examined and identified as the absence of RE (requirement engineering process). Despite all these results, efforts have not been put to avoid those problems that are becoming the cause of failure of projects that are outsourced i.e. RE process (Iqbal, Ahmad, Nizam, Nasir, & Noor, 2013).

Thus, it is important to improve RE practices as it has potential to reduce the development cost, delivering software on time as well as ensure the quality of software. In order to have a better RE practices, it is important to understand and assess the current practice implemented by the software developers. The Information Technology UUMIT has developed various software to support UUM main business activity. Thus, it is important to extend a support by investigating how currently software developers practice RE during software development. In addition, based on the study that carried out by Tahir and Ahmad (2010), there is scarcity of studies which conducted on the requirements engineering practices in general and in UUMIT in particular. As well as, Basharat et al., (2013) stated that, identify the requirements engineering practices is crucial for any organization.

1.3 Research Questions

The following research questions are important to this study:

a) What are the current Requirements Engineering practices?

b) How the software developers at University Utara Malaysia Information Technology (UUMIT) conduct the requirements engineering practices during software development?

c) What Requirements Engineering practices should be conducted during software development at Information Technology (UUMIT)?

1.4 Research Objectives

The following objectives are important in order to answer the research questions:

a) To review the current Requirements Engineering practices.

b) To investigate how the software developers at University Utara Malaysia Information Technology (UUMIT) conduct Requirements Engineering practices (RD and RM) during software development.

c) To propose Requirements Engineering practices during software development at Information Technology (UUMIT).

1.5 Significance of Research

This study is significant as it shows the evidence of RE practices among software developers at University Utara Malaysia Information Technology UUMIT. Thus, this study is able to highlight which activity of requirements development and requirements management has been adequately conducted and which activity needs to be improved in the future software development. As a result, it can help the UUMIT to produce quality software within the development cost on time.

This study is significant to the management of University as this study shows that which activity needs improvement so that the management is able to allocate budget to provide adequate and precise training as well as seminar for the software developers. By having these appropriate training and seminars, the software developers are able to have knowledge in order to improve the RE activities and therefore could led to successfully developed quality software. Hence, this will reduce the software development cost, on time delivery without compromising the software quality.

1.6 Research Contribution

Software failure has been a critical issue in almost software industry. This is because software cannot be delivered in time, exceeds the budget and lacks of expectation quality. In order to minimize this software failure, RE plays a vital role in software project development. Theoretically, an effective RE practice is compulsory in every software project development in order to be successful in delivering high quality software. In order to be successful in software project development, this study has the following contributions:

- This study demonstrates the current RE practices among software developers at UUM Information Technology UUMIT. Indeed, this study is able to point out which activity in requirements development and requirements management needs to be given attention in order to improve the RE practices among the software developers

- The result of this study will benefit the management of Information Technology UUMIT in order to identify the most suitable RE tool to be employed during software development. Hence, the result of this study can be an evidence and justification to be presented to the top UUM Management in planning the cost for software development in the future.
- The result of this study is an initial evidence of how Information Technology UUMIT has produced the various software in order to support UUM main business activity. Thus, this study will help the UUM top management to allocate cost and resources for Information Technology UUMIT in the future.

1.7 Scope of the Study

This study focuses on investigating the RE practices among the software developers at UUM IT. This is because Information Technology UUMIT is a centre that is responsible for development of software in order to support UUM main business activities.

The RE practices are only focusing on the following activities:

i. Requirements description

ii. Requirements development

iii. Requirements management

Therefore, other areas in RE such as method, tools and others are out of this study scope and will not be investigated in this study.

CHAPTER TWO
LITERATURE REVIEW

2.1 Requirement

Every project has some basic requirements that define what the end users, customer, developers, suppliers or business that are stakeholder stakeholders require from it and also there are some need of the systems for efficient functioning. (Hull, Jackson &Dick,2011). Requirement is a key factor during every software development as it describes what different stakeholders need and how system will satisfy these needs. They are generally expressed in natural language for the reason that everyone can well understand it. It helps the analyst to better understand which element and function are necessary in the development of particular project.

Moreover, requirements are consider as an input to design, implementation and validation phase of software project development. Thus, a software project is successful or failure during software development because of poor requirement elicitation as well as in requirements managing process. A survey conducted by Basharat et al. (2013) shows identified that in many unproductive projects absence of RE is the major cause of failure of such projects in software industry. However, it has also been recognized that with the help of suitable requirement engineering practices subsidized in software development process can leads to the success of projects.

Brooks defined the requirement as

> "*The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as*

difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later." (Brooks & Bullet, 1987)

According to Pfleeger and Atlee (2006), requirements are categorized as functional, non-functional requirements and constrains.

2.1.1 Functional Requirements

Functional requirements are statement of services or functionality that system should provide and how the system should react in a particular situation. Functional requirements are the interactions between the system and its environment independent from implementation. Sometime functional requirements are also stated as system constraints (Lauesen, 2002). These requirements generally depend on user of the software and type of system being developed.

2.1.2 Non Functional Requirements

These are the constraints on the services or functions offered by the system such as timing, development process and standards. These requirements define system properties like reliability of the system, response time and storage requirements etc. Process requirements may also be specified mandate a particular system, programming language or development method. Non-functional requirements may be more crucial than functional requirements if these are not met, the system is useless. User visible aspects of the system not directly related to functional behavior. Also known as quality attributes of the sys (Lauesen, 2002).

2.1.3 Constraints

These are also known as Pseudo requirements, imposed by the client or the environment in which the system will operate. They can be input/output device capability, system representation in the environment (Lauesen, 2002).

2.2 Requirements Engineering

The development of Software Requirement engineering Method (SREM) in 1977 has revealed terms *requirements* and *engineering* by Alford (Alford, 1977). Primarily, RE was employed to information development, after that become important to organizational as well as software application.

Whenever the engineering discipline has been embedded into requirements, the research direction has been incorporated the engineering approach to traditionally system analysis, as a result RE research domain become prevalent.

Previously, RE practices are focusing at the beginning of software development phase, known as system analysis. In the 1990s, RE has been acknowledged because RE is a vital phase of every software development project. Since then, the boundary of RE has prolonged beyond the perception of system analysis. Presently, RE is a well-known discipline and RE domain range from methods, approaches, techniques and tools.

According to Zave and Jackson (1997), RE is one of software engineering branches and RE is the earliest phase of the software development life cycle. In addition, Zave and Jackson (1997) describes RE 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."

In broad context, the requirement engineering deals with not only technical issues but it also supports the managerial, organizational, economic, and social issues. It is used to design software that meets the goal for which it was intended. According to (Cheng & Atlee, 2009), RE begins by identifying the need of stakeholders, understanding the context in which the software will be employed, modeling, analyzing, negotiating, and documenting the stakeholders' requirements; evaluating that the documented requirements match the negotiated requirements; and managing requirements evolution. The end users, customers, decision makers and developers are involved in a requirement engineering process as stakeholders. These stakeholders are from various backgrounds and have many individual and organizational goals due to the environment in which they work. It is not easy to produce a complete, consistent and well-structured set of requirements from incomplete, imprecise and conflicting sources. So the requirement is incomplete without considering the physical and organizational environment in which the system will be used.

Cheng and Atlee (2009) in their study concluded that RE is a complicated process because of the challenges faced by RE communities are different from software engineering community because requirements are located at the problem area while software is located at solution space. Furthermore, RE concerns with describing the problems to make it clear but software engineering is focusing on determining and proposing the solution (Cheng & Atlee, 2009). RE process has a great impression on software quality because the most expensive, frequent and dangerous type of software errors are related to poor requirements (Young, 2004). These errors affect the cost of software development because the cost to correct these errors increases as the time delay in finding them. In view of such difficulties, requirement engineering process should be more disciplined.

2.3 The Significance of Requirement Engineering

RE is becoming essential and researchers are giving attention to RE. Many researchers underlined that RE activities mainly contributes to the software product quality based on empirical study as well as industrial evidence (Brooks & Bullet, 1987; El Emam & Birk, 2000; Hoare, 1981). This is also supported by the CHAOS report published by the Standish Group. In (Chaos, 1995), having established RE practices contributed more than 42% of overall project success. Likewise, inappropriate RE practices represent more than 43% of the reasons for software failure.

Furthermore, (The Standish Group International, 2001) affirmed that requirements are the main reason for software projects failure. Hence, it means that all requirements elicited from users and customers are not included in the final software product (Leffingwell & Widrig, 2003). In addition, many previous researchers have identified that 70% of the requirements were difficult to identify and 54% were not clear and well organize (Gause & Weinberg, 1989). Gause and Weinberg (1989) also pointed out:

i. Requirements are difficult and challenge to describe in natural language.

ii. Requirements have many different types and level of details.

iii. Requirements are difficult to manage if they are not in control.

iv. Most of the requirements change during software development.

Requirements are important during software development because it describes what the user needs and why it is needed. Thus, this is essential to identify the correct requirement as early as possible during software development because the cost of correcting an error that is pointed out later in a software project is higher than to fix it right after its construction (Grady, 1999). The major significances of requirements problem is rework. Rework takes 30 to 50 percent of overall development cost (Boehm, W, Papaccio, & N, 1988) as well as requirements error contributes to 79 to 85 percent of total rework cost (Leffingwell, 1997). This is affirmed by (Leffingwell & Widrig, 2003) in their study has shown that 40% of the requirements generate rework during the software development.

2.4 Requirement Engineering Practices

In these days, RE becomes important and getting attention from the software community. This is because having good RE practices reduce the development cost and time and increase the quality of the software product (El Emam & Madhavji, 1995). In a study by (Tahir & Ahmad, 2010) indicated that it is important to have good RE practices in order to ensure the software product is developed properly.

According to a study by Quispe, Marques, Silvestre, Ochoa, and Robbes (2010), having poor RE practices have severe impact to software products. They indicated that the having poor RE practices lead to the following problems:

- Rework – Whenever there are any requirements changes, this means that there are parts of the system need to be updated based on the changes. These changes significantly influent the development of the system and this will cause delay and rework.
- Communication and coordination problem – software developers usually manage their requirements using many resources, such as text document, spreadsheet, presentation slides, email messages and many others. Thus, this is difficult to get fast, right and accurate requirements that lead to communication and coordination problem.
- Poor visibility of project status – Most of the projects do not have any requirements-related metrics in order to guide the project successfully, avoid rework, scope creep and manage requirements changes during software development project. This lead to poor project visibility and pushes the project manager to decide based on uncertainty.

In addition, Quispe et al. (2010) concluded that all the problems mentioned above significantly impact the success rate of a software project. It can be seen that there is important for the organization to have established and good RE practices in order to ensure software could be able to deliver successfully within the developments cost and time, without compromising the software quality.

2.5 Requirements Engineering Domain

The are two sets of activities in requirements engineering as; software requirements development phase and software requirement management (s et al., 2014) while some authors named it as requirements management (Leffingwell & Widrig, 2000). Similarly, many researchers within the community suggested viewing RM as a part of RE (Bell, Morrey, & Pugh, 1997; Pressman, 2010; Sommervile & I, 2010). However, authors (K. E. Wiegers, 2003) splits the domain of software RE into requirements development (RD) and requirements management (RM), as shown in Figure 1 below. In this study, the domain of RE is adopted from (K. E. Wiegers, 2003) that consider RE domain is a combination of RD and RM.

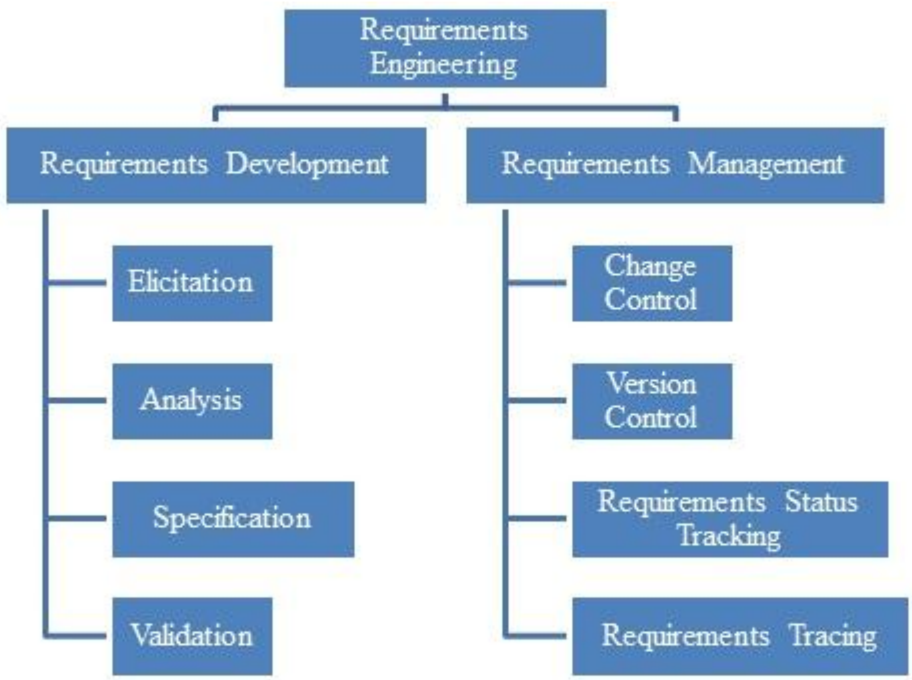

Figure 2.1. Requirements Engineering domain (Wiegers & Beatty, 2013; Wiegers, 2003)

2.5.1 Requirements Development

Requirements development is focusing on eliciting, analyzing, descripting and validating activities (Abran & Moore, 2001), as shown in Figure 2.1 above. It is a subdiscipline that encompasses all activities involved in elicitation, analysis, documentation and validation of requirements. The activities for the subcomponents of requirements development are (K. E. Wiegers, 2003):

i. Elicitation

- identifying the potential users
- gathering requirements from each user
- Understanding user task, goals and the business objectives in order to align them

ii. Analysis

- Analyzing the information elicited from users in order to determine the functional and non-functional requirements, constrains and rules
- Allocate the top-level requirements to software components
- Understanding the quality attributes
- Negotiate implementation significances

iii. Specification

- Translating the user requirements into requirements specifications and diagrams.

iv. Validation

- Review the requirements in order to understand the user requirements and identify any problems before implement it.

2.5.2 Requirements Management

Requirement management is an important phase in the requirement engineering process. RM is intentionally to manage the product's requirements, components and determining the conflict occurs in requirements, project plan as well as work product (Product Development Team, 2002b). The main aim is to ascertain an agreement of requirements and its meaning between customer and software team (Cuevas & Serrano, 2004; Product Development Team, 2002b) (Product Development Team, 2002a). Hence, RM is an activity that controls all information pertaining to requirements and everything that concern with the integrity of those requirements. RM involves (K. E. Wiegers, 2003):

- Manage changes in the requirements
- Handling the version of each requirement as well as the related documents.
- Managing and tracking the requirements' status
- Dealing with the link of each requirements towards the work product

2.6 Systematic Literature Review (SLR)

The researcher follows the approach proposed by Kitchenham (2004) for systematic literature review. A systematic review is a means of evaluating and interpreting all the available research that is relevant to a particular research question, topic area, or phenomenon of interest. It aims at presenting a fair evaluation of a research topic by using a trustworthy, rigorous, and auditable methodology.

A systematic review involves several stages and activities, which are briefly explained below: The primary studies are selected, the quality assessment used to include studies, the data extraction and monitoring is performed and the obtained data is synthesized. The conducting systematic literature review possesses the following activities: selection of primary studies, data extraction, methodology, results and findings, and conclusions. The SLR used also to identify the sources for data and how researcher collected data in RE researches. The primary studies were selected was described in the previous sections. Based on this information, the research resulted in

206 potentially related papers and thesis significant for the research topic. 35 papers and thesis have been found identical with the objectives of this study.

Table 2.1 presents the results of the final set of relevant papers selection for each of the sources in SLR. Searching results row shows the number of papers obtained from each source that resulted from the search string and the manual search. Finally selected row indicates the number of papers included to review after the rejection of papers that found irrelevant to this study. At least one of the exclusion criteria is applied to the systematic revision purpose. Duplicated papers were discarded, taking into consideration the large number of digital libraries available to students in Universiti Utara Malaysia, the newest publication, or the most complete one. And the thirdly, in this study, the researcher concentrate on the search terms (keywords) related to this research to answer the research questions, the key word (Requirement engineering practices).

Table 2.1

Systematic Literature Review of journal papers and thesis on RE practices

Source	IEEE	ACM	Emerald	Thesis	Springer	IJCET	IARIA
Searching results	115	30	26	10	20	2	3
Finally selected	11	11	5	5	1	1	1

The results from SLR shows that requirements engineering is defined by most of researchers as the process that develops product specifications that are complete, consistent and unambiguous. Scholars state that Requirement Engineering is considered the most imperative phase in software development process (Basharat *el at*, *2013*). The

RE practices have been considered as a key issue that affects the success rate of projects in software industry. The systematic literature reports a variety of Requirement engineering practices. It has been seen from previous a study that RE process plays an important role in providing successful software projects.

To summaries the systematic literature review, the table below will categories the finding based on the requirement engineering practices, as illustrates in table 2.2:

Table 2.2

Summarizes of systematic literature review

RE practices	RE phases
Description	The understanding of system requirements is critical to develop good systems.
Elicitation	Getting good requirements and effectively managing those requirements is a strong predictor of project success.
Requirement gathering and analysis	There exist obvious relationship between requirement gathering and analysis and software quality in requirement engineering process.
Validation	To validate must choose a group of experts from the case organizations.

Management	Project failures are necessarily not caused by technical issues, the cause is in general more likely related to the management, requirements and resources. Sustainable development, able to maintain a constant pace. Continuous attention to technical excellence and good design. Simplicity the art of maximizing the amount of work not done is essential Self-organizing teams. Good requirements engineering practices affects the software project success.
Agile	Many scholars recommended to apply requirements engineering practices to the agile development process. An agile software development team relies on communication and collaboration to perform requirements engineering activities. Developers have great satisfaction with regards to the efforts of RE practices applied in their agile projects. There is empirical evidence that requirements engineering practices correlated with Agile software development. Agile can help to achieve projects are built around motivated individuals, who should be trusted. Agile working software is the principal measure of progress in RE.

Methodology	The majority of studies used a mixed mode methodology (qualitative and quantitative). Using any software development methodology that includes RE processes will lead to better results. Most of studies uses quantitative analysis and survey, where some researchers use a combination of requirement engineering techniques for a real life complex IT project.
Outsourced	RE practices must be given due attention in order to improve the RE process for outsourced software development projects
Small , Medium and Large Enterprises	The practice of Requirements Engineering in Small and Medium Enterprises is weak and need for major development. Larger software development groups tend to have more well defined software development processes. Big organizations are more likely to have teams for software developing.
Face-to-face communication	Face-to-face conversation is the best form of communication (co-location) is important in RE.

2.7 Previous Work

In the previous literature review, there are numerous empirical studies on RE. In 1995, (El Emam & Madhavji, 1995) conducted a study of RE practices in the development of information system and they discovered the RE processes for information system based on the analysis of 60 cases and 7 technical/non-technical issues. Moreover, in 2007, (Sadraei, Aurum, Beydoun, & Paech, 2007) have conducted a RE survey in Australian companies. In 2007, (Aranda, Easterbrook, & Wilson, 2007) performed an empirical study on how small companies conducting their RE activity.

On the other hand, there is an research interest on how the globalized software engineering practices effect the RE practices. In 2004, (Damian, Zowghi, Vaidyanathasamy, & Pal, 2004) conducting a study in Australian organization and pointed out several key factors that influent the success of requirements management in software development structure. In addition, a study by (Niazi & Babar, 2007) that conducting a survey at Vietnamese software industry has concluded that cultural issues in maintaining trust in outsourcing software. Further surveys also been conducted by (Hickey & Davis, 2003; Sim, Alspaugh, & Al-Ani, 2008) who concentrates on requirements expert. Their surveys were focus on specific problem rather than understanding the general industry problem.

In 2010, L. Liu, Li, and Peng (2010) performed an empirical study of RE in Chinese companies. In this study, they reported that RE was failed in the industry and they provided the reasons why it was failed. In the Malaysian software industry, a recent study was conducted in 2014 by (Rahman, Haron, Sahibuddin, & Harun, 2014). In this study, they focus on the RE practice in Malaysia public sector. This study reported that the main problem is the communication between system analyst and stakeholders.

In the same context Quispe, Marques, Silvestre, Ochoa and Robbes (2010) found that, there is a lack of knowledge about the requirements engineering practices in these types of very small software companies. The results of their study are listed in the following points:

- Project specifications are usually met, but the client often finds the solution unsatisfactory; this leads to the conclusion that
- communication issues with clients cause incomplete specifications
- the project's scope expands as clients require additional changes, often with inadequate changes
- this ad-hoc process leads to requirement management issues such as loss of requirements
- when uncertainty arises, developers tend to resolve the issue without contacting the clients.

  In addition, the researchers used quantitative approach, 24 experimented project managers from different companies are participated in the study. Data gathering was performed through two instruments: a survey and a focus group.

In 2010, Savolainen, Kuusela and Vilavaara used descriptive method to understanding how Applied requirements engineering practices to the agile development process in two industrial cases. They investigated the transition to Agile Development Rediscovery of Important Requirements Engineering Practices. The researchers found that, agile development in this context needs very skilled developers and has to be a combination of new and old practices. They also found that it is difficult for large organizations to be agile. Work allocation for a large number of different teams with different competencies tends to decrease speed, and increase the role of design and management. Trying to add completely new practices on top of agile development can do more harm than good. One can easily end up remove those properties of the agile practices that may make them successful.

In sum up, based on the above studies it is obvious that the main activities in requirements engineering process such as feasibility study, requirements prioritization, having standard template for requirements, using software systems for managing requirements, identifying non-functional requirements are well practiced by the surveyed firms. More than 50% of the firms always or often practiced.

Other study, Abdullah, Honiden, Sharp, Nuseibeh and Notkin (2011) designed an ethnographically informed approach. This approach aims to study practice without interference, treats all data. The agile development team studied is based in a large telecommunications and media company in the UK; the observation took place in the team's offices and lasted for four days. They present an initial study of a commercial agile steam and their communication patterns that support RE. Evidence from practice indicates that two simple physical artefacts (story cards and the wall), used in a particular and disciplined manner, and supported by appropriate social activity, are key to the success of co-located agile teams. Their study also shows that an agile software development team relies on communication and collaboration to perform requirements engineering activities.

Moreover, in 2012, Alnafoosi conducted a survey on the current landscape of telecommunication industry interaction with Requirement Engineering. The participants were IT employees working telecommunication companies. The findings show the importance of aspects of change, complexity, internal and external components present a challenge in requirement engineering to capture and present early stable requirements. The result of this study shows that in many cases, the situation of conflicting software development requirements which is complicated further by the interaction with existing systems. That can create large, complex, and conflicting requirements.

On the other hand, Mahrin (2013) identified situation factors in requirement engineering practices, which are in the following:

- PROCESS: Tasks, Techniques, Process Selection, Process Maturity Tools, Management Knowledge, Management Performance, Management Decision.
- PROJECT: Problem, Domain, Project Characteristics, Requirement, Evolution, Project Goals, Requirements Estimation, Project Risks, Project Requirements.

It is evident that there is a lack to situational factors affecting the Requirement Engineering process during Global Software Development is presently not available. The lack of such study is challenging as it not only restrain the ability to improve Requirement Engineering process.

Jebreen and Wellington (2013) used a mixed mode methodology (Quantitative & Qualitative). The data was collected throughout field work in two Jordanian multinational Software Development businesses, with a combined workforce of approximately 40 employees and consider as medium size software producer companies.

Total Number of interviewees equal 30. The findings of his study show that in a packaged software implementation, the software company has substantial control over the development of packaged software, and the client organization becomes vulnerable to software company actions. Future research could consider problematic area within existing RE tools is that they do not support a distributed collaboratively collection and analysis of requirements, which can be said is necessary in the packaged software context since packaged software requirements for implementation comes from defendable requirements. Moreover a packaged software implementation is a unique type of IS software, with characteristics that distinguish it from requirement engineering, traditional system development and initial adoption of a commercial system.

While Iqbal, Ahmad,Nizam, Nasir and Noor (2013) used a different methodology based on Systematic Literature Review in addition to a quantitative methodology by investigating RE practices in 15 renowned software companies of Pakistan we conducted our study in several stages. The questionnaire was sent to more than 20 companies through an email. Out of those we have got 15 responses. The results of this study shows that there activities differ from businesses to businesses and organization to organization. The cultural and geographical aspects also affect these activities. Small and medium organizations practice the main re activities like feasibility study, requirements analysis and prioritization but usually ignore the more subtle activities like conflict resolution, requirement inspections and risk analysis. This study focuses to which extent these practices are being followed in small and medium software industry of Pakistan. The study also tries to identify the potential limitations to adopt appropriate requirements engineering practices. Some possible solutions are also provided in this study.

Therefore based on the evidences above, the researchers conclude that requirement engineering is considered the most imperative phase in software development process. The RE practices have been defined as a key issue that affects the success rate of projects in software industry. This conclusion is identical with the findings from a study conducted by Kassab, (2014) who used a quantitative methodology and survey responses that captured a diverse mix of industries, job roles, and project domains and RE practices. Respondents were asked to base all their project responses on one project only that they were either currently involved with or had taken part in during the past five years. Participants of the survey were drawn from multiple Sources, where 247 participants from 23 countries (with a completion rate of 48%). The result of Mohamad's study show a number of RE practices show no significant difference between agile and waterfall Requirements Validation. The Overall respondents expressed greater satisfaction with regards to the efforts of RE practices applied in their agile projects compared to the waterfall projects. In addition to that Mohamad's study reported on the current landscape of practice in requirements elicitation, analysis and presentation, management, effort estimation and tools with respect to agile methodologies.

Daneva and Ahituv (2010) pointed that the practices to the levels of coordination complexity, in a way that converged with RE practices. The study also indicated implications of the findings of our group study for future research. The study evaluates 12 practices for engineering the coordination requirements in inter-organizational

Enterprise, Resource Planning projects, Focus Froup (FG) methodology, in generally, a FG is a group discussion on a given topic, which is monitored, facilitated and recorded by a researcher. In essence, the researcher provides the focus of the discussion, and the data comes from the group interaction. A set of 12 practices for engineering the coordination requirements in an (ERP) project.

Bjarnason, Wnuk and Regnell (2011) used qualitative approach and case study to collect empirical data from industrial projects at a large company that is using a product line approach. The practitioners from a large software development company, the result shows that Agile practices address some RE challenges such as communication gaps and over scoping, but also cause new challenges, such as striking a good balance between agility and stability, and ensuring sufficient competence in cross-functional development teams.

Other study conducted by Where Liskin, Schneider, Fagerholm, and Münch, (2014) used direct observation, surveying and interviews to gain information on the requirements artifacts that were used in the project described in the previous section. The qualitative analysis was used to gain insights into the role of the artifacts. Researchers present the detailed research questions and our procedures for data collection and analysis. The participants are students filled out a questionnaire where they entered each story card they had worked on and the time that they worked on it. Furthermore, the students stated whether the duration had corresponded with what they had implicitly expected or whether the implementation of the task took longer or shorter time. The found that the experience of programmer played an important role in the project. Although the participants felt that their communication was good, the project still suffered from misunderstandings. The study found Good requirements communication is crucial to the success of agile projects.

Recently, Unterkalmsteiner, Feldt and Gorschek, (2014) propose a definition of requirements engineering and software test (REST) alignment, a taxonomy that characterizes the methods linking the respective areas, and a process to assess alignment. IT developed an assessment framework (REST-bench), applied it in an industrial assessment. Applying an in-depth cases and illustrate angles of analysis on a set of thirteen alignment methods. They an assessment framework (REST-bench), applied it in an industrial assessment, and showed that it, with a low effort, can identify opportunities to improve REST alignment. In the same vein, Ralph and Kelly (2014) produced several findings as stating in the following:

Design work is largely organized into projects

- Designers are often fixated on an explicit client
- Designers appear more concerned with being on time than adhering to budgets or contracts
- The aesthetic connotation of design is not universal
- Designers desire interesting work and capable colleagues
- Designers are more concerned with clients' emotional reactions to their products than with satisfying explicit requirements
- Designers perceive analysis and design as closely related

- Designers recognize that contracts, plans, schedules and budgets are often unreasonable or misguided.

To reach the above findings Ralph and Kelly used qualitative methodology and analysed transcripts of interviews with diverse design professionals including software engineers. The researcher used a semi-automated content analysis technique to derive 11 themes from the corpus. The interview transcripts were analysed using semi-automated content analysis. This analysis begins with unsupervised semantic mapping. An interdisciplinary sample of 191 design professionals (68 in the software industry) was interviewed concerning their perceptions of success. Non-software designers (e.g. architects) were included to increase the breadth of ideas and facilitate comparative analysis.

In 2014 also, Rahul et al., found that the understanding of system requirements is critical to develop good systems. Participants who received requirements framing produced significantly less original designs than participants who received ideas framing.

As well as, Jain, Cao, Mohan and Ramesh (2014) adopted a multiple-case study methodology in four organizations, and a qualitative methodology. Through multisite case studies, the researchers were able to compare practices across conventional and agile environment. The primary source of data was semi-structured interviews with individual informants at Alpha, Gamma, and Delta. Senior managers in the focal organizations helped the researchers to identify informants who were involved in RE. Data was collected from four sites through interviews, emails, observations, and other documentation. Total number of interviewees equal to 45, interviews has been conducted with the following personnel:

(1) Project managers should attempt to balance the needs of domain and application engineering teams.

(2) Project managers should be cognizant of the need to tailor boundary spanning practices to address RE challenges based on the development approach (agile vs. conventional).

(3) Project managers who designate boundary spanners and specify the types of boundary objects need to recognize the complementarity of boundary spanner roles and boundary objects.

(4) Project team members should be cognizant of the nature of knowledge and knowledge processes that are supported by the boundary spanning practices.

The researcher argues that the framework developed in this study provides the much needed theoretical foundation to develop large-scale empirical studies of the role of boundary spanning in RE. It is shown that Requirements Engineering (RE) faces considerable challenges that are often related to boundaries between various stakeholders involved in the software development process.

Moreover, Aedah (2014) evaluated RE practices by analysing the progress and RE processes applied on the three real-time and embedded software projects, A, B and C. Aedah found that Good requirements engineering practices affects the software project success rate. Also she concludes that RE approach for embedded software projects must be capable of evolving with the dynamic changing user needs such as from errors and incomplete requirements and many influencing factors in the real time and embedded applications domain. The researcher uses determination of process improvement (PI) method, selection of product-based software projects and evaluation of the PI method.

Based on the above it is evident that the primary success of a software system is the degree it meets the purpose for which was intended. Therefore, software systems requirements engineering (RE) helps to achieve this purpose.

In the same context Kabaale and Kituyi (2015) used qualitative research approach was adapted. Four software companies in Uganda were purposively selected to participate in the study. Data were collected using questionnaires. The requirements for designing the framework were gathered and refined from both primary and secondary data. Four case organizations in Uganda were selected. A total of 60 respondents were purposively selected from these four institutions to participate in the study.

The study show that the key requirements for process improvement in small and medium software companies were identified as user involvement, use of evolutionary requirements engineering process improvement (REPI) strategy, change management, training and education, management support and commitment. The designed framework was validated to ensure that it can be applied in RE and process improvement in small and medium software companies. Validation results show that the proposed framework is applicable and can be used to improve RE and process improvement in small and medium software companies.

The situation of RE and process improvement is even worse in small and medium enterprises that produce software. Consequently, the quality of software being produced by these companies has kept deteriorating. The purpose of this paper is to design a framework that will help small and medium software companies improve their RE processes in order to compete favorably with larger software companies, more especially in terms of software quality.

Lindholm (2012) suggest that requirements engineering should be a process driven practice, and the same should in general apply to any kind of project work, with a clearly defined and structured process plan which is meant to be followed. The later an issue is detected or a request for some kind of a change is made during a software engineering or development project, the more it will take time and resources to fix or implement it. Project failures are necessarily not caused by technical issues, the cause is in general more likely related to the management, requirements and resources.

Talbot (2011) uses both quantitative and qualitative. A range of data collection methods employed in primary research. These include critical incident technique, diaries, focus groups, interviews, the observation, protocol analysis and questionnaires. A survey questionnaire instrument is designed and deployed. In this type of research the Respondent is asked to reflect on specific circumstances, giving information as to what happened and what the result was. The responses showed that in 25% of the companies who have RE specialists, they have no formal training in the discipline. The number of companies reporting that they have one or more employees whose primary responsibility is RE was 67%.

Pandey, Suman, and Ramani (2010) concluded that the relationship between explicit and implicit activities and project success is important to identify the nature of activities in software development project and the success of RE process improvement. They used a qualitative methodology, where a structured interview was the main tool for data collection. The researcher examined many RE models to find out the good requirement for development of quality software product. Interviewees from two Indian companies. The number of interviewees was 80. RE process awareness is low in small IT projects. The role of RE was influenced by the size of the project large projects allowed for greater separation of tasks between the roles of business analyst and project manager.

Other study was by Niazi, El-Attar, Usman and Ikram (2012) who identified 11 frequently cited high value RE practices that should be planned and implemented in projects to avoid frequently occurring requirements related problems.

CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Introduction

The methodology for this study is adapted from (Mohamed, Farvin, Baharom, & Deraman, 2013) in their study regarding current software development practices from agile perspective. The methodology has three main phases which are:

- Phase 1: Theoretical study.
- Phase 2: A survey of RE practices at Information Technology (UUMIT).
- Phase 3: Mapping the best RE practices with current RE practices conducted at Information Technology (UUMIT).
- Figure 3.1 below indicates the research framework for this study.

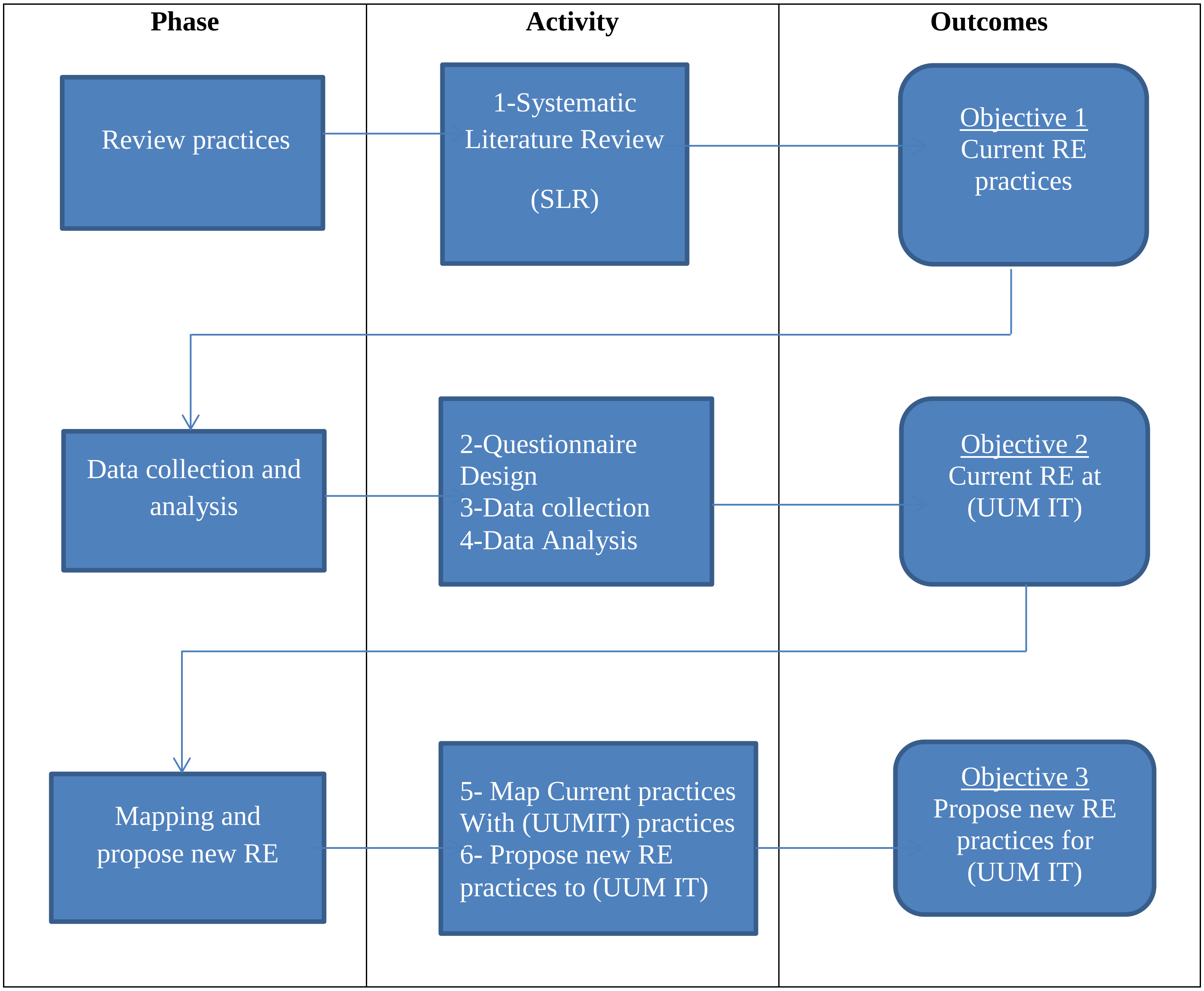

Figure 3.1. The Research Framework

3.2 Phase 1: Theoretical study

In this phase, the aim is to conduct a review of current RE practices. Thus, systematic literature review (SLR) was conducted in order to have a complete review of RE practices. The best RE practices will be identified and this will be the platform in order to propose good RE practices for Information Technology UUMIT.This lead to achieve the first objective of this study.

3.3 Phase 2: A survey of RE practices at Information Technology (UUMIT)

In this phase, a survey was conducted at Information Technology UUMIT. The aim of this survey is to investigate how the software developers at the UUM IT conduct their RE practices presently. The objectives of this survey are:

- To investigate how the software developers describe their requirements.
- To investigate how the software developers conduct the requirements development during software development.
- To investigate how the software developers conduct the requirements management during software development.

3.3.1 Instrument Design

For this current study, a self-administered questionnaire is being used to collect the data regarding current RE practices at UUM IT. The questionnaire is adapted from Anderson and Felici (2001); Tahir and Ahmad (2010); Iqbal et al. (2013), and Khankaew and Riddle (2014). As shown in the table 3.1:

Table 3.1

Questionnaires sources

Activity(s)	Items	Sources
Description	**A.1** Do you have standards templates / documents for describing requirements?	Anderson and Felici (2001)
	A.2 Do you have a specific lay out for the requirements document to improve readability?	Zainol,Mansoor (2008)
	A.3 Do you have guidelines how to write requirements?	
	A.4 Do you produce a summary of the requirements?	Tahir and Ahmad (2010)
	A.5 Do you make a business case for a system?	
	A.6 Do you have a glossary of specialized terms?	Niazi, El-Attar, Usman & Ikram, (2012)
	A.7 Is the requirements document easy to change?	
	A.8 Do you use diagrams appropriately?	Iqbal,Ahmad,Nizam,Nasir & Noor,(2013)
	A.9 Do you supplement natural language with other descriptions of requirements?	
	A.10 Do you specify requirements quantitatively?	

Elicitation	**BA.1** Do you carry out a feasibility study before starting a new project? **BA.2** While eliciting requirements are you sensible to organizational and political factors which influence requirements sources? **BA.3** Do you use business concerns to drive requirements elicitation? **BA.4** Do you prototype poorly understood requirements? **BA.5** Do you use scenarios to elicit requirements? **BA.6** Do you define operational processes? **BA.7** Do you reuse requirements from other systems which have been developed in the same application area?	Anderson and Felici (2001) Massimo Felici (2004) Tahir and Ahmad (2010) Besrour and Ghani (2012) Niazi, El-Attar, Usman & Ikram, (2012) Iqbal,Ahmad,Nizam,Nasir & Noor,(2013) Khankaew and Riddle (2014)
Analysis and Negotiation	**BB.1** Do you define system boundaries? **BB.2** Do you use checklists for requirements analysis? **BB.3** Do you encourage the use of electronic systems **BB.4** (e.g., e-mail) to support requirements negotiations? **BB.5** Do you plan for conflicts and conflict resolution? **BB.6** Do you prioritise requirements? **BB.7** Do you classify requirements using a multidimensional? **BB.8** approach which identifies specific types **BB.9** (e.g., hardware-software, changeable-stable, etc.)? **BB.10** Do you use interaction matrices to find conflicts and overlaps? **BB.11** Do you perform any risk analysis on requirements?	Anderson and Felici (2001) Massimo Felici (2004) Tahir and Ahmad (2010) Besrour and Ghani (2012) Niazi, El-Attar, Usman & Ikram, (2012) Iqbal,Ahmad,Nizam,Nasir & Noor,(2013) Khankaew and Riddle (2014)

Validation	**BC.1** Do you check that requirements document meets your standards?	
	BC.2 Do you organize formal requirements inspections?	Anderson and Felici (2001)
	BC.3 Do you use multi-disciplinary teams to review requirements?	
	BC.4 Do you involve external (from the project) reviewers in the validation process?	Massimo Felici (2004)
	BC.5 In order to focus the validation process do you define validation checklists?	Niazi, El-Attar, Usman & Ikram, (2012)
	BC.6 Do you use prototyping to animate / demonstrate requirements for validation?	Iqbal,Ahmad,Nizam,Nasir & Noor,(2013)
	BC.7 Do you propose requirements test cases?	Khankaew and Riddle (2014)
	BC.8 Do you allow different stakeholders to participate in requirements validation?	
Management	**C.1** Do you uniquely identify each requirement?	
	C.2 Do you have defined policies for requirements management?	Anderson and Felici (2001)
	C.3 Do you record requirements traceability from original sources?	Massimo Felici (2004)
	C.4 Do you define traceability policies?	Zainol,Mansoor (2008)
	C.5 Do you maintain traceability manual?	Tahir and Ahmad (2010)
	C.6 Do you use a database to manage requirements?	Besrour and Ghani (2012)
	C.7 Do you define change management policies?	Niazi, El-Attar, Usman & Ikram, (2012)
	C.8 Do you identify global system requirements?	
	C.9 Do you identify volatile requirements?	Hamid, Raza and Naqvi(2013)
	C.10 Do you record rejected requirements?	Iqbal,Ahmad,Nizam,Nasir & Noor,(2013)
	C.11 Do you reuse requirements over different projects?	Khankaew and Riddle (2014)

The data will be collected to achieve the underlying research objectives for the current study. The adopted questionnaire consists of (49) questions, which comprises of two sections; first section the demographic information for the UUMIT staff. The second section consists of the required engineering practices. The items are measured on a five likert scale, as adopted from Zainol and Mansoor (2008); Tarawneh, Baharom, Yahaya and Zainol (2011). The detailed illustration about the structure for the questionnaire is given below:

- Part 1 - Demographic Profile
- Part 2 - Current RE Practices

    - Functional and non-functional requirements
    - Requirements development
    - Requirements management

In order to ensure that the questions that are included in the questionnaire are able to collect information that will achieve the survey objective, it is important to map each activity with questions to achieve objective 2. Thus, the figures below (Figure 3.2, 3.3 and 3.4) summarize this.

Figure 3.2. The mapping each activity with the questions

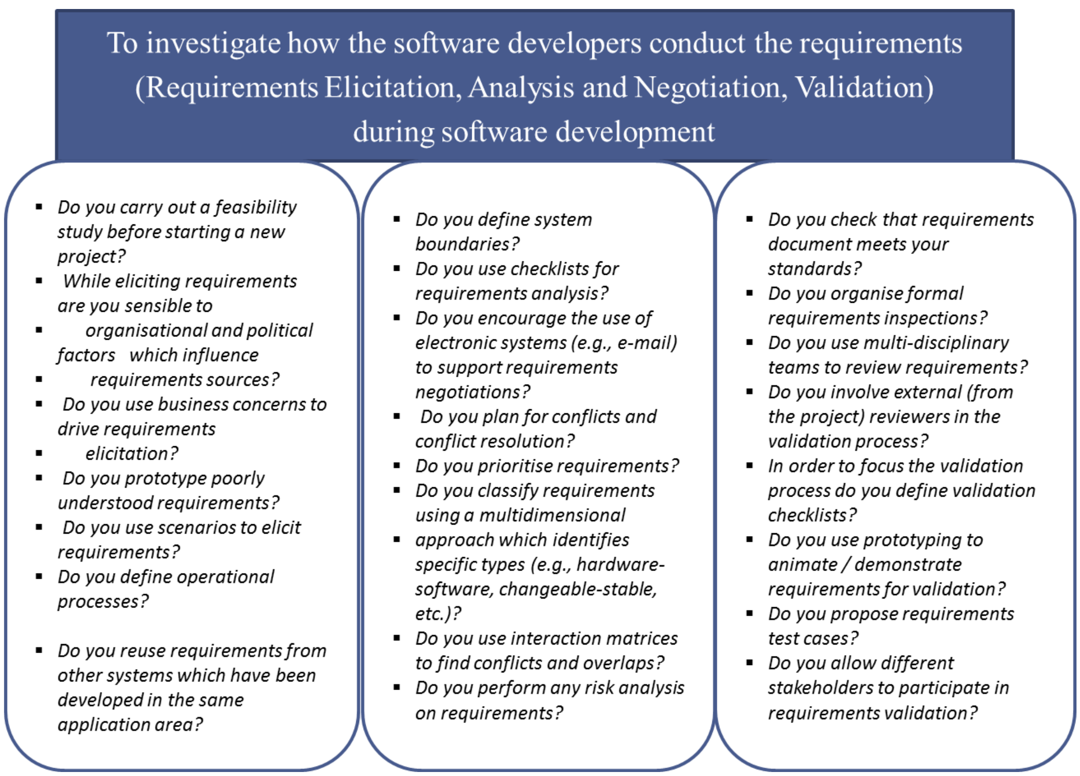

Figure 3.3. The mapping each activity with the questions

Figure 3.4. The mapping each activity with the questions

3.3.2 Data Collection

In this phase, the data was collected using a set of questionnaire. The questionnaire was distributed to all software developers at Information Technology UUMIT.

3.3.3 Data Analysis

The data collected during data collection phase was analyzed in this phase. The data was analyzed using statistical analysis. Thus, SPSS software was used to analyze data statistically. Thus, this is to ensure the second objective of this study will be obtained.

3.3.4 Phase 3: Mapping The Best RE Practices With Current RE Practices Conducted At Information Technology (UUMIT).

In this phase, it is important to map the current RE practices from objective 1 with the current RE practices conducted at the UUMIT. This is important in order to guarantee that software developers are alert and prepared to employ good RE practices. Thus, a detailed analysis will be conducted and the result of this analysis will propose good RE practices for software developers during software development project at UUM IT. Therefore, the third objective will be achieved.

CHAPTER FOUR
FINDINGS

4.1 Introduction

This chapter presents the analysis of primary data collected from the 20 participants in the survey of this study. The analysis in this chapter aims to achieve objective of the study to measure the current RE at UUMIT as described in second phase of research methodology Mohamed et al.(2013)The researcher collected data from participants in the survey of RE practices at UUMIT In this phase . The aim of the survey is to investigate how the software developers at the UUMIT conduct their RE practices presently.

The analysis includes the following variables:

1. Requirements Description (A)
2. Requirements Development (B), which compose the following variables:
- Requirements Elicitation (BA)
- Requirements Analysis and Negotiation (BB)
- Requirements Validation (BC)
3. Requirement Management (C)

A self-administered questionnaire is used to collect the data regarding current RE practices at UUM IT for each of variable. The questionnaire consists of (49) questions in two sections; first include the demography information for the UUM IT staff. Whilst, the second section includes the items of the each variables associated with requirement engineering practices. The analysis of each variable is divided into two sections as in the following:

- **The first section** is descriptive analysis, where the researcher uses descriptive statistics (mean, standard deviation, frequencies) to understand the relationship between the variables and the correlation between the items of each variable. The analysis uses charts and frequencies to describes the responses of participants on each item, and make the conclusion on the significance of each item.
- **The second section** of analysis uses factor analysis which includes (correlation matrix, rotation matrix, scree plote, KMO and Bartlett's Test).
- **Analysis Categories**

The analysis of each variable will be conducted in three categories (Reliability coefficient, Descriptive Statistics and factor analysis)

4.2 Descriptive Statistics

Descriptive statistics is the discipline of quantitatively describing the main features of a collection of information (Mann& Prem S, 1995). The descriptive analysis applied in this section to evaluate the variables of the study and measure the relationship of these variables from the descriptive analysis. The results of this analysis are used to validate the hypotheses of the study. Each item in the questionnaire analyzed separately using the following descriptive statistics:

1. Mean
2. Standard Deviation

3. Frequencies

The descriptive statistics is used in this study in order to identify the significance of the constructs (items) of each factor (latent variable). After data has been entered using SPSS software (V.20), data were analyzed using descriptive statistics. The main reason for using descriptive analysis it is because this type of analysis is commonly used for summarizing data frequency or measures of central tendency (mean, standard deviation, frequencies).

- **Mean.** In probability and statistics, mean and expected value are used synonymously to refer to one measure of the central tendency either of a probability distribution or of the random variable characterized by that distribution (Feller & William, 1950). The mean, median and mode are all estimates of where the "middle" of a set of data is. These values are useful when creating groups or bins to organize larger sets of data.
- **Standard deviation.** The standard deviation is the average distance between the actual data and the mean. In probability and statistics, mean and expected value are used synonymously to refer to one measure of the central tendency either of a probability distribution or of the random variable (Feller & William, 1950).

It is a measure that is used to quantify the amount of variation or dispersion of a set of data values (Bland, Martin, Altman & G, 1996). A low standard deviation indicates that the data points tend to be very close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values. A standard deviation close to 0 indicates that the data points tend to be very close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values see (Figure 4.1 below).

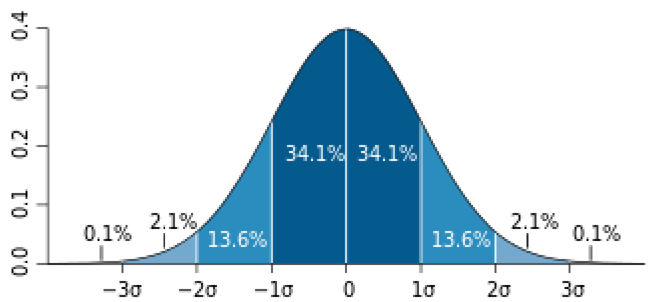

Figure 4.1. Standard deviation of normal distribution data

- **Frequencies.** The frequency of an event ί the number $\eta_{ί}$ of times the event occurred in an experiment or study (Kenney & Keeping, 1962). These frequencies are often graphically represented in histograms see (Figure 4.2 below)**.** A frequency distribution table is an arrangement of the values that one or more variables take in a sample. Each entry in the table contains the frequency or count of the occurrences of values within a particular group or interval, and in this way, the table summarizes the distribution of values in the sample. The researcher uses SPSS frequency analysis in order to shows the number of occurrences of each response chosen by the respondents.

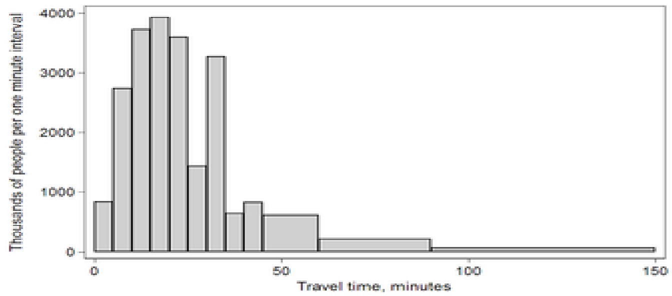

Figure 4.2. Frequencies of observations occurring in certain ranges of values

- **Percentages.** Cumulative percentage is another way of expressing frequency distribution. It calculates the percentage of the cumulative frequency within each interval, much as relative frequency distribution calculates the percentage of frequency. The main advantage of cumulative percentage over cumulative frequency as a measure of frequency distribution is that it provides an easier way to compare different sets of data. Cumulative frequency and cumulative percentage graphs are exactly the same, with the exception of the vertical axis scale. In fact, it is possible to have the two vertical axes, (one for cumulative frequency and another for cumulative percentage), on the same graph (Stangor & Charles, 2010).

4.3 Factor Analysis

Factor analysis is a multivariate statistical technique, which is used for the resolution of a set of a large number of variables in terms of relatively few hypothetical variables, called factors (Rummel, 2007; Shenoy & Madan Pant, 1994). Such analysis is also used to find ways of condensing that information which is contained in a number of original values into only a few dimensions.

According to Shenoy et al. (1994), factor analysis results serves three main purposes:

1) To identify the underlying, or latent, factors which determine the relationship between observed variables.

2) To clarify the relationship between the variables.

3) To provide a classification scheme, in terms of which data scores on various rating scales are grouped together.

In this study the researcher will use confirmatory factor analysis, and anticipating that there are a number of different variables underlying certain dimensions, whereas others belong to other dimensions, uses factor analysis to confirm such an assumption.

4.4 Demographic Analysis

The researcher collected the demographic data of 20 participants to make analysis on the demographic characteristics of the study sample. In the following (Table 4.1), the percentages, mean and frequencies of participants' demographic data is divided into four categories:

Table 4.1

The demographic data of study sample

Demographic Variable	Category	Frequencies	Percentage	Mean	Standard Deviation
Experience Years	1-5 years	6	30.0	2.20	1.05
	6-10 years	7	35.0		
	11-19 years	4	20.0		
	> 20 years	3	15.0		
5 past Years achievement	1-5	10	50.0	1.55	0.60
	6-10	9	45.0		
	11-19	1	5.0		
Current Position at UUMIT	Assistant officer	8	40.0	1.7	0.65
	IT officer	10	25.0		
	IT executive	2	10.0		
Period length in current position	1 to 5	12	60.0	1.65	0.93
	6 to 10	4	20.0		
	11 to 19	3	15.0		
	>= 20	1	5.0		

4.4.1 Experience Years

The results show that the majority of participants having years of experience range 6-10 years (35%), where the mean value shows that the average experience range between 6-10 years, while the standard deviation value shows that other participants cluster away from the mean with high difference, in other words there is a big differences in experience between the staffs of UUMIT. The lowest experience years is for those experienced more than 20 years, only 3 employees in UUMIT have long experience (15%). Where those having lowest experience (1-5 years) are high and represent 30% of the total number of staff. The chart below shows the distribution of data for experience years.

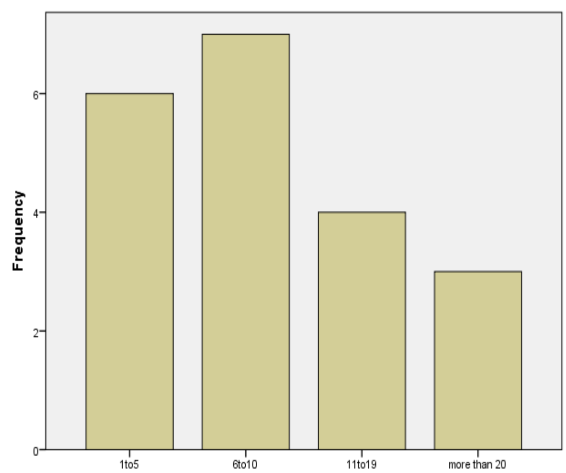

Figure 4.3. Experience years

4.4.2 Achievement of Five Years:

The results show that the majority of participant achieved 1-5 software projects (50 %), where the next percentage those who achieved 6-10 projects (45%). It means that the majority of staffs in UUMIT achieved software projects 1-10. Only one staff has achieved high number of projects 11-19. This result shows the need for major improvement in the current achievements by the staff of UUMIT to increase the number more than 10 projects. The mean value equal 1.55 which means that the majority of IT staffs cluster between first and second selection and cluster very close to the mean confirmed by very small standard deviation (0.60). The chart below shows the distribution of data for achievement of five years.

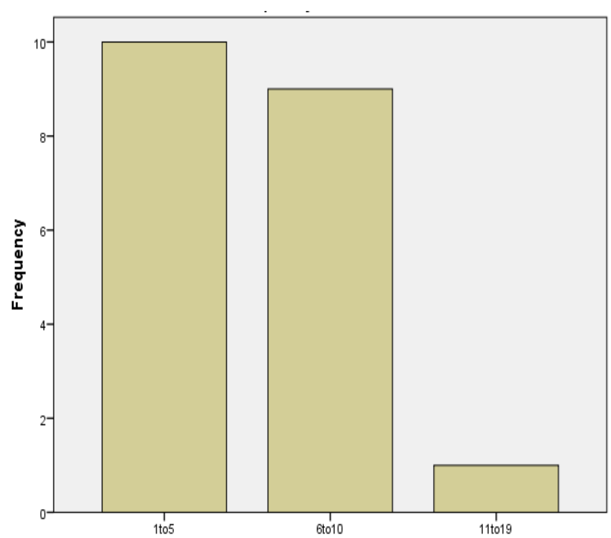

Figure 4.4. Achievement of five years

4.4.3 Current Position at UUMIT

The results show that the majority of participants are IT officers (50%), the next are assistant officers (40%). The lowest number is IT executive (10%). This result shows that the current positions of staffs in UUMIT are encouraging and the number of employees divided between experienced staff (IT officers) and fresh experience (assistant officers). Also it is found that only one or two IT executives available in the department, the researcher suggest assigning more IT executives to work in IT department in order to increase the quality of achievements and improve the overall performance. The chart below shows the distribution of data for current position at UUMIT.

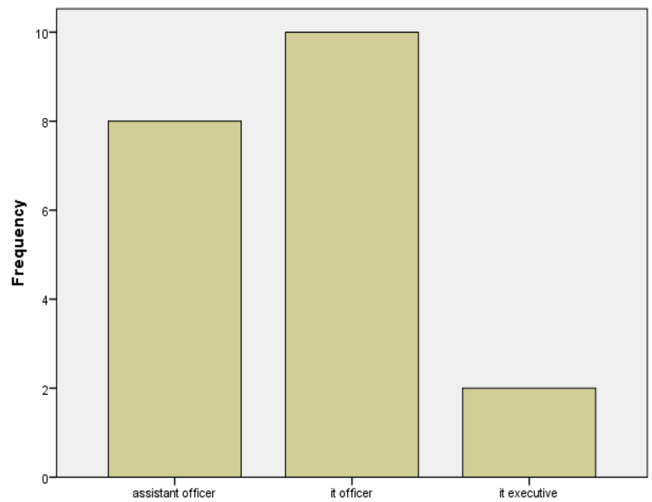

Figure 4.5. Current position at UUMIT

4.4.4 Period Length In Current Position

The result shows that the majority of participants (60%) have stayed in their current position at UUMIT between 1-5 years. While only few number of staffs stayed in their current position between 6-10 years (20%), and the rest (15%) stayed more than 11 years, where only one staff having very long experience (>20 year) in working with UUMIT. The current result show the need to encourage the staffs to work in the department be providing incentives in their work and encourage teamwork so that more staff stay to in IT department. The mean equal 1.65 which confirms the conclusion on period length in current position at UUMIT. The chart below shows the distribution of data for this characteristic.

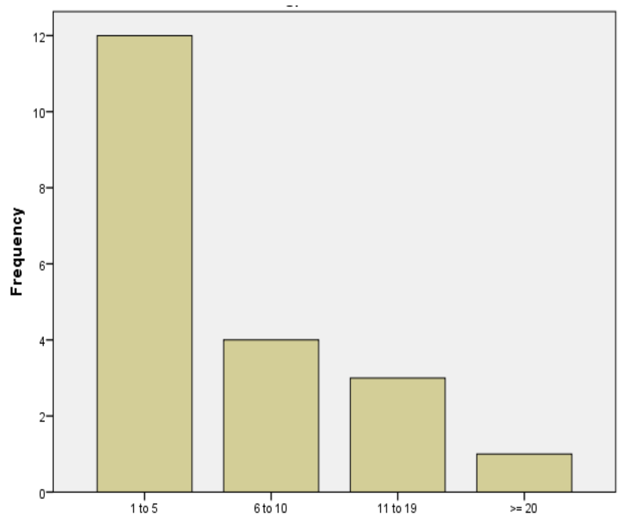

Figure 4.6. Period length in current position

4.5 Reliability Coefficient

Reliability is used in this study to assess the internal consistency of each factor and the overall all data. In internal consistency reliability estimation is used for single measurement instrument administered to a group of people on one occasion to estimate reliability.

In effect the researcher judges the reliability of the questionnaire by estimating how well the items that reflect the same construct yield similar results (Joseph et al., 2003). According to that the researcher is looking at how consistent the results are for different items for the same construct within the measure. There are a wide variety of internal consistency measures that can be used. In this study the researcher uses Cronbach's Alpha coefficient.

Cronbach's alpha reliability coefficient normally ranges between 0 and 1. However, there is actually no lower limit to the coefficient. The closer Cronbach's alpha coefficient is to 1.0 the greater the internal consistency of the items in the scale (George, 2003).

The reliability coefficient Cronbach's Alpha in this study equal to 0.954 (95.40%), which is a high value and shows that the reliability of collected data is high and can be used for factor analysis (table.4.2).

Table 4.2

Reliability Statistic

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.954	.957	45

The following table shows the reliability coefficient of each variable.

Table 4.3

Reliability coefficient of RE variables

Variable name	**Reliability coefficient (Cronbach's Alpha)**
Requirements description (A)	0.929
Requirements Elicitation (BA)	0.773
Requirements Analysis and Negotiation (BB)	0.917
Requirements Validation (BC)	0.858
Requirements Management (C)	0.841

4.6 Factor Analysis of Requirement Description

1. Correlation Matrix

The correlation matrix shows a significant correlation and greater than 0.7 between the following items:

- Items (1,2)
- Items (1,3)
- Items (1,4)
- Items (1,6)
- Items (1,8)
- Items (1,10)
- Items (2,6)
- Items (2,8)
- Items (3,4)
- Items (7,8)

The above correlations are considered high and show strong relationships between the items of this variable. The highest correlation is between item 7 and item 8 (0.910). This correlation shows that the participants have common opinion with regards to item 7 (the requirements document easy to change) and item 8 (using diagrams appropriately), and both requirements document and diagrams are important in requirement description stage. The overall result is good and shows strong and moderate correlation between all items. No negative correlation is found.

Table 4.4

Correlation matrix of requirement description

Item	1	2	3	4	5	6	7	8	9	10
1	1.000	.916	.725	.774	.325	.788	.638	.736	.573	.721
2	.916	1.000	.637	.742	.435	.792	.589	.743	.445	.609
3	.725	.637	1.000	.873	.223	.531	.386	.507	.514	.349
4	.774	.742	.873	1.000	.319	.566	.484	.604	.619	.440
5	.325	.435	.223	.319	1.000	.410	.500	.560	.062	.278
6	.788	.792	.531	.566	.410	1.000	.435	.596	.471	.501
7	.638	.589	.386	.484	.500	.435	1.000	.910	.543	.466
8	.736	.743	.507	.604	.560	.596	.910	1.000	.556	.564
9	.573	.445	.514	.619	.062	.471	.543	.556	1.000	.592
10	.721	.609	.349	.440	.278	.501	.466	.564	.592	1.000

a. Determinant = 8.121E-006

2. Rotated Component Matrix

The table below shows that only 5 items with eigenvalues greater than 0.7 are loaded on first component. These items are (4, 3, 1, 9, 2). Other items are loaded with low eigenvalue or on component 2.

Table 4.5

Rotated Component Matrix

Rotated Component Matrix[a]	Component	
	1	2
A.4 Do you produce a summary of the requirements?	.868	.215
A.3 Do you have guidelines how to write requirements?	.852	.431
A.1 Do you have standards templates / documents for describing requirements?	.846	
A.9 Do you supplement natural language with other descriptions of requirements?	.752	.133
A.2 Do you have a specific lay out for the requirements document to improve readability?	.735	.522
A.6 Do you have a glossary of specialized terms?	.654	.443
A.10 Do you specify requirements quantitatively?	.593	.389
A.5 Do you make a business case for a system?		.863
A.8 Do you use diagrams appropriately?	.516	.778
A.7 Is the requirements document easy to change?	.392	.767

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.
a. Rotation converged in 3 iterations.

3. KMO and Bartlett's Test

The KMO and Bartlett's Test for this factor shows that KMO value equal 0.615. This value is considered good and moderate, also the significance = 0.000, which is less than 0.05. This result shows that the data of this factor is highly fit for factor analysis.

Table 4.6

KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.615
Bartlett's Test of Sphericity	Approx. Chi-Square	173.862
	df	45
	Sig.	.000

4. Scree Plot

The Scree Plot shows the underlying components and the eigenvalues of each principal component on the y-axis (the underlying values eigenvalue) and number of principal component on the x-axis , and through the chart below, it is shown that the items (4, 3, 1, 9, 2) are highly loaded on the first component.

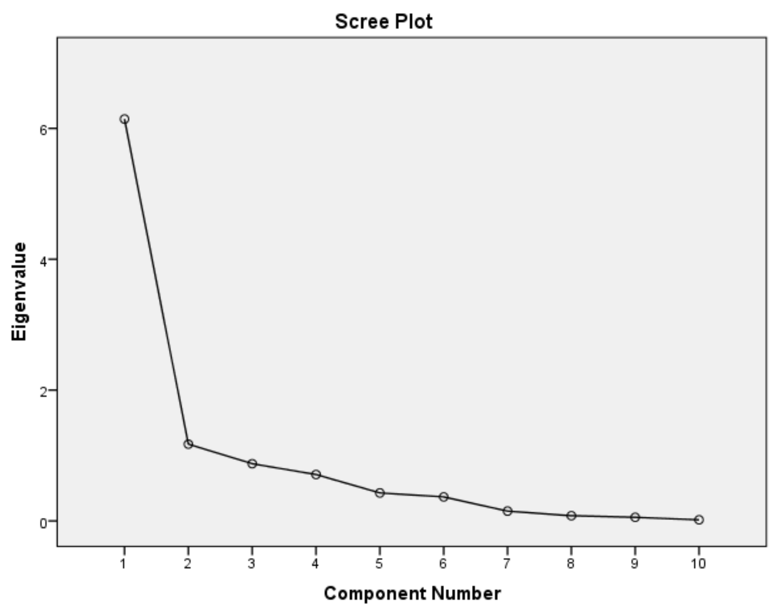

Figure 4.7. The loading of items on the principal components

4.7 Factor Analysis of Requirements Elicitation

1. Correlation Matrix

The below correlations matrix shows moderate relationships between the majority of items. The highest correlation is between item 5 and item 6 (0.639). This correlation shows that the participants have common opinion with regards to item 5 (using scenarios to elicit requirements) and item 6 (defining operational processes), and both requirements are important in requirement development stage. The overall result is good and shows strong and moderate correlation between all items. No negative correlations are found.

Correlation Matrix

Table 4.7

Correlation matrix of requirement elicitation

Item	1	2	3	4	5	6	7
1	1.000	.321	.671	-.068	.473	.466	.423
2	.321	1.000	.395	.533	.308	.020	.163
3	.671	.395	1.000	.160	.450	.400	.204
4	-.068	.533	.160	1.000	.564	.139	.447
5	.473	.308	.450	.564	1.000	.639	.619
6	.466	.020	.400	.139	.639	1.000	.276
7	.423	.163	.204	.447	.619	.276	1.000

a. Determinant = .022

2. Rotated Component Matrix

The table shows that only 3 items with eigenvalues greater than 0.7 are loaded on first component. These items are (1, 3, and 6). Other items are loaded with low eigenvalue or on component 2.

Table 4.8

Rotated Component Matrix

Rotated Component Matrix[a]	Component	
	1	2
BA.1 Do you carry out a feasibility study before starting a new project?	.888	
BA.3 Do you use business concerns to drive requirements elicitation?	.761	.185
BA.6 Do you define operational processes?	.749	.114
BA.4 Do you prototype poorly understood requirements?		.949
BA.2 While eliciting requirements are you sensible to organizational and political factors which influence requirements sources?	.136	.670
BA.5 Do you use scenarios to elicit requirements?	.610	.637
BA.7 Do you reuse requirements from other systems which have been developed in the same application area?	.388	.582

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.
a. Rotation converged in 3 iterations.

3. KMO and Bartlett's Test

Table 4.9

KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.546
Bartlett's Test of Sphericity	Approx. Chi-Square	60.773
	df	21
	Sig.	.000

The KMO and Bartlett's Test for this factor shows that KMO value equal 0.546. This value is considered acceptable, also the significance = 0.000, which is less than 0.05. This result shows that the data of this factor is fit for factor analysis.

4. Scree Plot

The Scree Plot shows the underlying components and the eigenvalues of each principal component on the y-axis (the underlying values eigenvalue) and number of principal component on the x-axis , and through the chart below, it is shown that the items (1, 3, and 6) are highly loaded on the first component.

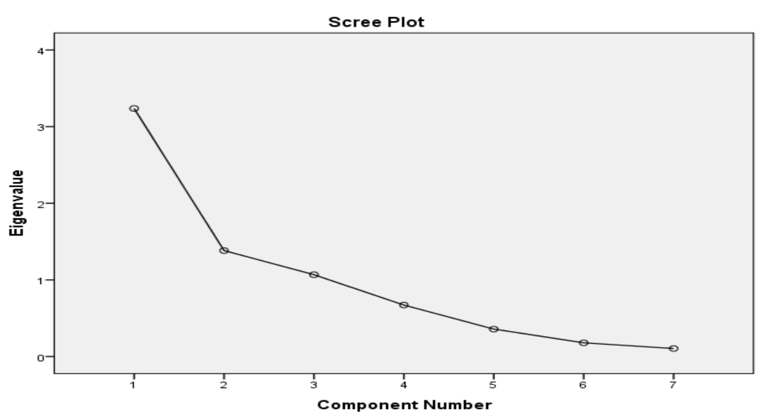

*Figure 4.8.*The loading of items on the principal components

4.8 Factor Analysis of Requirements Analysis and Negotiation

1. Correlation Matrix

The below correlations matrix shows moderate relationships between the majority of items. The highest correlation is between item 1 and item 6 (0.780). This correlation shows that the participants have common opinion with regards to item 1 (defining system boundaries) and item 6 (classification of requirements using a multidimensional), and both these two requirements are important in requirement development stage. The overall result is good and shows strong and moderate correlation between all items associated with Requirements Analysis and Negotiation. No negative correlations are found.

Table 4.10

Correlation matrix of Requirements Analysis and Negotiation

Item	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	1.000	.390	.742	.616	.379	.780	.511	.640	.571
2	.390	1.000	.471	.642	.275	.393	.472	.455	.450
3	.742	.471	1.000	.676	.639	.629	.590	.391	.598
4	.616	.642	.676	1.000	.573	.662	.646	.700	.690
5	.379	.275	.639	.573	1.000	.418	.409	.260	.611
6	.780	.393	.629	.662	.418	1.000	.526	.583	.571
7	.511	.472	.590	.646	.409	.526	1.000	.697	.745
8	.640	.455	.391	.700	.260	.583	.697	1.000	.636
9	.571	.450	.598	.690	.611	.571	.745	.636	1.000

a. Determinant = .022

2. Rotated Component Matrix

The table shows that only 3 items with eigenvalues greater than 0.7 are loaded on first component. These items are (8, 7, and 4). Other items are loaded with low eigenvalue or on component 2.

Table 4.11

Rotated Component Matrix

Rotated Component Matrix[a]	Component	
	1	2
BB.8 Do you use interaction matrices to find conflicts and overlaps?	.908	
BB.7.Approach which identifies specific types	.755	.332
BB.4.Do you plan for conflicts and conflict resolution?	.731	.503
BB1. Do you define system boundaries?	.665	.464
BB.2.Do you use checklists for requirements analysis?	.662	.180
BB.6.Do you classify requirements using a multidimensional?	.650	.466
BB.9 Do you perform any risk analysis on requirements?	.633	.556
BB.5.Do you prioritise requirements?	.107	.918
BB.3.Do you encourage the use of electronic systems (e.g., e-mail) to support requirements negotiations?	.433	.782

3. KMO and Bartlett's Test

The KMO and Bartlett's Test for this factor shows that KMO value equal 0.772. This value is good, also the significance = 0.000, which is less than 0.05. This result shows that the data of this factor is strongly fit for factor analysis.

Table 4. 12

KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.772
Bartlett's Test of Sphericity	Approx. Chi-Square	109.254
	df	36
	Sig.	.000

4. Scree Plot

The Scree Plot shows the underlying components and the eigenvalues of each principal component on the y-axis (the underlying values eigenvalue) and number of principal component on the x-axis , and through the chart below, it is shown that the items (8, 7, and 4) are highly loaded on the first component.

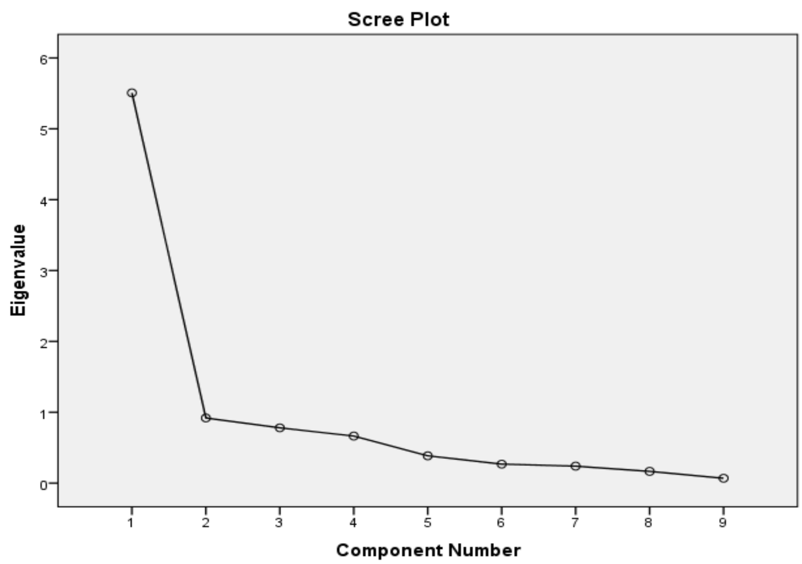

Figure 4.9. The loading of items on the principal components

4.9 Factor Analysis of Requirements Validation

1. Correlation Matrix

The correlation matrix shows a significant correlation and greater than 0.7 between the following items:

- Items (1,2)
- Items (4,6)
- Items (4,8)
- Items (6,8)

The below correlations matrix shows moderate relationships between the majority of items. No negative correlations are found. The highest correlation is between item 6 and item 8 (0.846). This correlation shows that the participants have close opinion with regards to item 6 (using prototyping to animate/demonstrate requirements for validation) and item 8 (allowing different stakeholders to participate in requirements validation), and both these two requirements are found important in requirement development stage. The overall result is good and shows strong and moderate correlation between all items.

Table 4.13

Correlation matrix of Requirements Validation

Item	1	2	3	4	5	6	7	8
1	1.000	.740	.249	.357	.602	.456	.273	.307
2	.740	1.000	.067	.311	.350	.248	.079	.164
3	.249	.067	1.000	.439	.305	.347	.416	.442
4	.357	.311	.439	1.000	.466	.757	.604	.757
5	.602	.350	.305	.466	1.000	.532	.368	.216
6	.456	.248	.347	.757	.532	1.000	.586	.846
7	.273	.079	.416	.604	.368	.586	1.000	.672
8	.307	.164	.442	.757	.216	.846	.672	1.000

a. Determinant = .022

2. Rotated Component Matrix

The table shows that 4 items with eigenvalues greater than 0.7 are loaded on first component. These items are (8, 6, 4 and 7). Other items are loaded with low eigenvalue or on component 2.

Table 4.14

Rotated Component Matrix

Rotated Component Matrix[a]

	Component	
	1	2
BC.8 Do you allow different stakeholders to participate in requirements validation?	.907	
BC.6 Do you use prototyping to animate/demonstrate requirements for validation?	.831	.328
BC.4 Do you involve external (from the project) reviewers in the validation process?	.826	.293
BC.7 Do you propose requirements test cases?	.812	
BC.3 Do you use multi-disciplinary teams to review requirements?	.606	
BC.1 Do you check that requirements document meets your standards?	.210	.907
BC.2 Do you organize formal requirements inspections?		.884
BC.5 In order to focus the validation process do you define validation checklists?	.366	.658

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.
a. Rotation converged in 3 iterations.

3. KMO and Bartlett's Test

The KMO and Bartlett's Test for this factor shows that KMO value equal 0.658. This value is accepted, also the significance = 0.000, which is less than 0.05. This result shows that the data of this factor is strongly fit for factor analysis.

Table 4.15

KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.658
Bartlett's Test of Sphericity	Approx. Chi-Square	88.016
	df	28
	Sig.	.000

4. Scree Plot

The Scree Plot shows the underlying components and the eigenvalues of each principal component on the y-axis (the underlying values eigenvalue) and number of principal component on the x-axis , and through the chart below, it is shown that the items (8, 6, 4 and 7) are highly loaded on the first component.

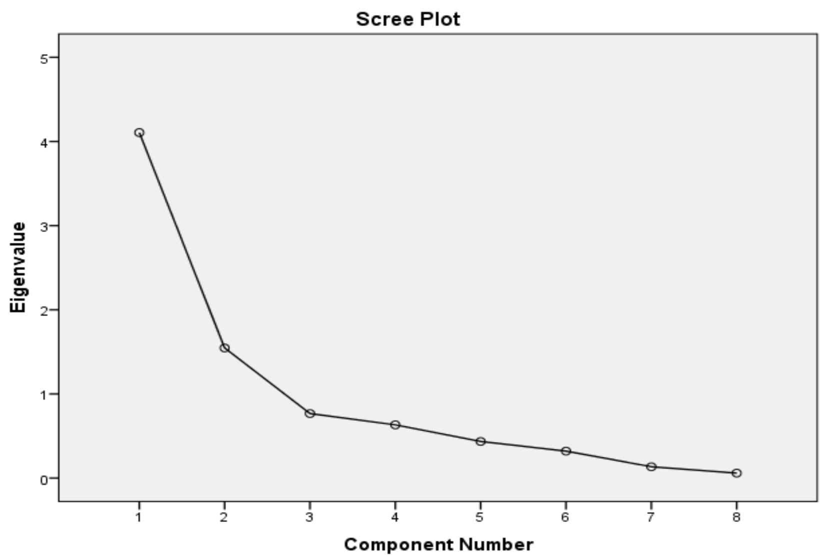

Figure 4.10. The loading of items on the principal components

4.10 Factor Analysis of Requirements Management

1. Correlation Matrix

The below correlations matrix shows a strong correlation between the majority of items. Where, the highest of the correlation between items (2, 4), (7, 8), (8, 12), and (9, 8), negative correlations are found. The overall result is good and shows strong and moderate correlation between all items.

Table 4.16

Correlation matrix of Requirements Management

Item	1	2	3	4	5	6	7	8	10	11	12
1	1.000	.769	.586	.562	.569	.027	.118	.146	.345	.110	.473
2	.769	1.000	.605	.792	.508	.355	.205	.343	.328	.353	.618
3	.586	.605	1.000	.631	.476	-.038	-.009	.097	.179	.237	.380
4	.562	.792	.631	1.000	.598	.212	.214	.169	.251	.353	.453
5	.569	.508	.476	.598	1.000	-.066	-.091	-.238	.182	.069	.000
6	.027	.355	-.038	.212	-.066	1.000	.084	.425	.100	-.234	.094
7	.118	.205	-.009	.214	-.091	.084	1.000	.730	.631	.414	.601
8	.146	.343	.097	.169	-.238	.425	.730	1.000	.785	.433	.748
9	.345	.328	.179	.251	.182	.100	.631	.785	1.000	.414	.664

10	.110	.353	.237	.353	.069	-.234	.414	.433	.414	1.000	.681
11	.473	.618	.380	.453	.000	.094	.601	.748	.664	.681	1.000

a. Determinant = .022

2. Rotated Component Matrix

The table shows that 4 items with eigenvalues greater than 0.7 are loaded on first component. These items are (8, 11, 7 and 9). Other items are loaded with low eigenvalue or on component 2.

Table 4.17

Rotated Component Matrix

Rotated Component Matrix[a]

	Component	
	1	2
C.8 Do you identify global system requirements?	.950	
C.11 Do you reuse requirements over different projects?	.854	.339
C.7 Do you define change management policies?	.829	
C.9 Do you identify volatile requirements?	.807	.172
C.10 Do you record rejected requirements?	.615	.202
C.6 Do you use a database to manage requirements?	.235	
C.2 Do you have defined policies for requirements management?	.373	.835
C.4 Do you define traceability policies?	.245	.832
C.1 Do you uniquely identify each requirement?	.186	.818
C.5 Do you maintain traceability manual?	-.184	.801
C.3 Do you record requirements traceability from original sources?		.791

3. KMO and Bartlett’s Test

The KMO and Bartlett's Test for this factor shows that KMO value equal 0.547. This value is accepted, also the significance = 0.000, which is less than 0.05. This result shows that the data of this factor is strongly fit for factor analysis.

Table 4.18

KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.547
Bartlett's Test of Sphericity	Approx. Chi-Square	159.770
	df	55
	Sig.	.000

4. Scree Plot

The Scree Plot shows the underlying components and the eigenvalues of each principal component on the y-axis (the underlying values eigenvalue) and number of principal component on the x-axis , and through the chart below, it is shown that the items (8, 11, 7 and 9) are highly loaded on the first component.

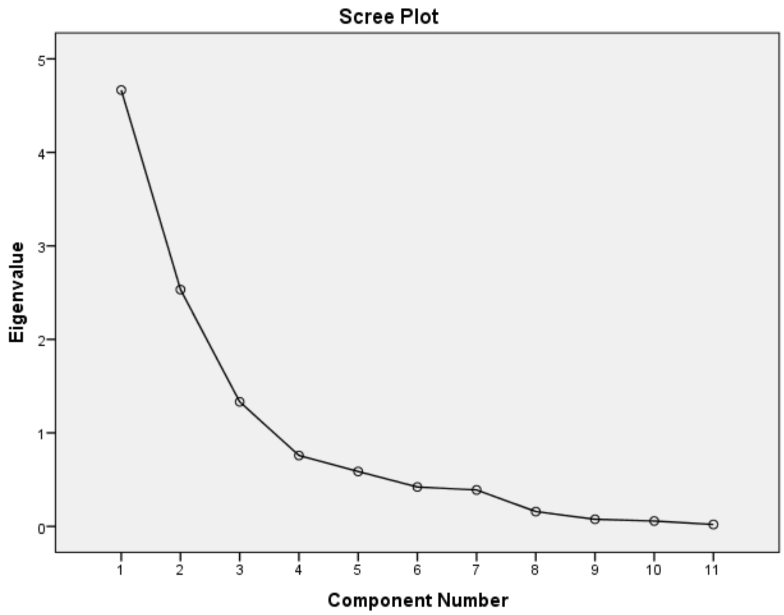

Figure 4.11. The loading of items on the principal components

CHAPTER FIVE
RESULTS AND DISCUSSION

5.1 Introduction

This chapter shows the results of the study. The study reviewed the current Requirement Engineering practices, and investigated how the software developers at University Utara Malaysia Information Technology (UUMIT) conduct Requirements Engineering practices (RD and RM) during software development. The study also conducted a systematic literature review (SLR) in order to identify the success factors of RE practices. The researcher investigated three variables; Requirements Description, Requirements Development (Requirements Elicitation, Requirements Analysis and

Negotiation, Requirements Validation), and Requirement Management. In the following section, the summary and conclusion for each variable is described.

5.2 Requirements Description

5.2.1 Descriptive Analysis of Requirement Description

This section analyzes the primary data using the basic descriptive analysis associated with requirement description. The basic descriptive analysis includes (Mean, Standard deviation). The overall results and responses on this factor are illustrated below:

Table: 5.1

Descriptive result of Requirement Description

Item	Description	Mean	St. Deviation
1	Do you have standards templates / documents for describing requirements?	4.05	.945
2	Do you have a specific lay out for the requirements document to improve readability?	4.00	.973
3	Do you have guidelines how to write requirements?	3.85	.933
4	Do you produce a summary of the requirements?	3.85	.875

5	Do you make a business case for a system?	3.65	.745
6	Do you have a glossary of specialized terms?	3.45	.887
7	Is the requirements document easy to change?	3.55	.826
8	Do you use diagrams appropriately?	3.50	.946
9	Do you supplement natural language with other descriptions of requirements?	3.25	.851
10	Do you specify requirements quantitatively?	3.50	.889

The values of mean and standard deviations of each item of requirement description are shown above. The descriptive results show a moderate agreement on all items, except item (1, and 2) where the participants conform that they have a standards templates/documents for describing requirements and they have a specific lay out for the requirements document to improve readability, the mean value of item 1 and 2 equal (4.05, 4.00) respectively. This means that the practice of requirement description by IT staff on items 1 and 2 is always, while other items are practices on regular bases but not always.

The weakest practice of requirement description is scored in item 9, where participants confirm that supplement natural language with other descriptions of requirements is achieved regularly but with low percentage as the standard deviation is high (.851) it means other opinions are cluster away from the mean value of this item.

The table(5.2) shows the frequencies of the item, the highest percentage of participants responses equal 55% on regular practice of this item, The result shows that the majority of participants practice this item regularly and it shows that IT staff in UMMIT have common agreement to practice this items on regular bases.

Table 5.2

Do you have standards templates / documents for describing requirements

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	never	1	5.0	5.0	5.0
	sometimes	2	10.0	10.0	15.0
	regularly	11	55.0	55.0	70.0
	always	6	30.0	30.0	100.0
	Total	20	100.0	100.0	

This is holistically presented in appendix C all the frequencies and charts for each items requirement description.

Requirement description is specified in this study as the initial step of analysis. The goal of requirement description is to get an overview of the raw requirement to the extent of it being understood by the IT Manager performing the continuous requirements engineering.

The results of the study show a moderate practices to requirement description among the staffs of UUMIT. It is found that the staffs have a frequent practice with in using standards templates/documents for describing requirements and they have a specific lay out for the requirements document to improve readability. The study shows that currently there is a weak practice in supplement natural language with other descriptions of requirements, which is done regularly and very often. In summary the staffs of UUMIT practice the following steps of requirement description on regular bases:

- Using guidelines to write requirements
- Producing a summary of the requirements
- Making a business case for a system

- Define a glossary of specialized terms
- Using diagrams appropriately
- Specifying requirements quantitatively

5.3 Requirements Development

Requirement development is associated with examining and gathering particular requirements and objectives for the system by the IT staff from customer, users, system's operating environment, and marketing and standard). The study investigated requirement development practice in UUMIT department. Three variables compose the requirement development are examined in the survey (Requirements Elicitation, Requirements Analysis and Negotiation, Requirements Validation).

5.3.1 Descriptive Analysis of Requirements Elicitation

This section analyzes the primary data using the basic descriptive analysis associated with Requirements Elicitation. The basic descriptive analysis includes (Mean, Standard deviation). The overall results and responses on this factor are illustrated below:

Table: 5.3

Descriptive result of Requirements Elicitation

Item	Description	Mean	St. Deviation
1	Do you carry out a feasibility study before starting a new project?	4.00	.649
2	While eliciting requirements are you sensible to organizational and political factors which influence	3.45	.759

	requirements sources?		
3	Do you use business concerns to drive requirements elicitation?	3.95	.605
4	Do you prototype poorly understood requirements?	3.20	1.196
5	Do you use scenarios to elicit requirements?	3.55	.686
6	Do you define operational processes?	3.20	.696
7	Do you reuse requirements from other systems which have been developed in the same application area?	3.80	.768

The values of mean and standard deviations of each item of Requirements Elicitation are shown above. The descriptive results show that the majority of participants do a regular practice of Requirements Elicitation, the conclusion is examined by the answers of participants on items (2-7) except item (1) which scored highest mean value (4.00) where the participants conform that they carry out a feasibility study before starting a new project always. The weakest practice of requirement development is found in item (4) and (6) which they scored (3.20, 3.20) respectively.

The table(5.4) shows the frequencies of the item, the highest percentage of participants responses equal 60% on regular practice of this item, The result shows that the majority of participants practice this item regularly and it shows that IT staff in UMMIT have common agreement to practice this items on regular bases. See more in Appendix C.

Table: 5.4
Do you carry out a feasibility study before starting a new project

	Frequency	Percent	Valid Percent	Cumulative Percent

Valid	sometimes	4	20.0	20.0	20.0
	regularly	12	60.0	60.0	80.0
	always	4	20.0	20.0	100.0
	Total	20	100.0	100.0	

See more in Appendix C.

Requirements Elicitation

The results of the study show that the majority of participants do a regular practice of requirements elicitation, IT staffs carry out a feasibility study before starting a new project always comparing to other practices. The weakest practice of requirement elicitation is doing prototype to understood requirements, and defining operational processes. In summary the staffs of UUM IT practice the following steps of requirements elicitation on regular bases:

- Staffs are sensible to organizational and political factors which influence requirements sources
- Using business concerns to drive requirements elicitation
- Using scenarios to elicit requirements
- Reusing requirements from other systems which have been developed in the same application area.

5.3.2 Descriptive Analysis of Requirements Analysis and Negotiation

This section analyzes the primary data using the basic descriptive analysis associated with Requirements Analysis and Negotiation. The basic descriptive analysis includes (Mean, Standard deviation). The overall results and responses on this factor are illustrated below:

Table: 5.5

Descriptive result of Requirements Analysis and Negotiation

Item	Description	Mean	St. Deviation
1	Do you define system boundaries?	4.20	.696
2	Do you use checklists for requirements analysis?	3.90	.852
3	Do you encourage the use of electronic systems (e.g., e-mail) to support requirements negotiations?	4.00	.918
4	Do you plan for conflicts and conflict resolution?	3.85	.933
5	Do you prioritize requirements?	4.10	.718
6	Do you classify requirements using a multidimensional?	3.90	.912
7	Approach which identifies specific types	3.60	.681
8	Do you use interaction matrices to find conflicts and overlaps?	3.70	.733
9	Do you perform any risk analysis on requirements?	3.80	.768

The values of mean and standard deviations of each item of Requirements Analysis and Negotiation are shown above. The descriptive results show that the majority of participants do a regular practice of Requirements Analysis and Negotiation. The majority of participants conform that they carry out the steps of Requirements Analysis and Negotiation on regular bases. Only item (1), (3) and (5) scored the highest mean value (4.20, 4.00, 4.10) respectively, which shows that IT staffs encourage the use of electronic systems to support requirements negotiations and define system boundaries always.

The table(5.6) shows the frequencies of the item, the highest percentage of participants responses equal 50% on regular practice of this item, The result shows that the majority of participants practice this item regularly and it shows that IT staff in UMMIT have common agreement to practice this items on regular bases.

Table: 5.6

Do you define system boundaries

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	sometimes	3	15.0	15.0	15.0
	regularly	10	50.0	50.0	65.0
	always	7	35.0	35.0	100.0
	Total	20	100.0	100.0	

See more in Appendix C.

Requirements Analysis and Negotiation

The results of the study show that the majority of participants do a regular practice of requirements analysis and negotiation as part of requirement development. Two practices found practiced always which they are encouraging the use of electronic systems to support requirements negotiations and defining system boundaries, these two practices always achieved by the staffs in UUM IT. The following requirements analysis and negotiation are practiced on regular bases:

- Checking lists for requirements analysis
- Planning for conflicts and conflict resolution
- Classifying requirements using a multidimensional
- Approach which identifies specific types
- Using interaction matrices to find conflicts and overlaps
- Performing any risk analysis on requirements

5.3.3 Descriptive Analysis of Requirements Validation

This section analyzes the primary data using the basic descriptive analysis associated with Requirements Validation. The basic descriptive analysis includes (Mean, Standard deviation). The overall results and responses on this factor are illustrated below:

Table: 5.7

Descriptive result of Requirements Validation

Item	Description	Mean	St. Deviation
1	Do you check that requirements document meets your standards?	4.10	.852
2	Do you organize formal requirements inspections?	3.95	.759
3	Do you use multi-disciplinary teams to review requirements?	3.65	.671
4	Do you involve external (from the project) reviewers in the validation process?	3.65	.813
5	In order to focus the validation process do you define validation checklists?	4.05	.605
6	Do you use prototyping to animate/demonstrate requirements for validation?	3.95	.826
7	Do you propose requirements test cases?	3.85	.745
8	Do you allow different stakeholders to participate in requirements validation?	3.95	.826

The values of mean and standard deviations of each item of Requirements Validation are shown above. The majority of participants conform that they carry out the steps of Requirements Validation on regular bases The descriptive results show that the majority of participants do a regular practice of Requirements Validation except item (1) and (5) that scored the highest mean value (4.10, 4.05) respectively. This result shows that IT staffs always check that requirements document that meets their standards, and define validation checklists always. All other opinions confirms high level of practice to Requirements Validation in IT department of UUM regularly.

The table(5.8) shows the frequencies of the item, the highest percentage of participants responses equal 40% on always practice of this item, The result shows that the majority of participants practice this item always and it shows that IT staff in UMMIT have common agreement to practice this items on always.

Table: 5.8
Do you check that requirements document meets your standards

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	sometimes	6	30.0	30.0	30.0
	regularly	6	30.0	30.0	60.0
	always	8	40.0	40.0	100.0
	Total	20	100.0	100.0	

See more in Appendix C.

Requirements Validation

The majority of staffs in UUM IT department confirm that they carry out the steps of requirements validation on regular bases, only checking that requirements document to verify meeting project standards, also IT staffs focus on the validation process checklists many often. While other practices of requirement validations are found satisfactory and achieved by staffs of UUM IT on regular bases. The following these practices:

- Organizing formal requirements inspections
- Using multi-disciplinary teams to review requirements
- Involving external (from the project) reviewers in the validation process
- Using prototyping to animate/demonstrate requirements for validation
- Proposing requirements test cases
- Allowing different stakeholders to participate in requirements validation

5.3.4 Descriptive Analysis of Requirements Management

This section analyzes the primary data using the basic descriptive analysis associated with Requirements Management. The basic descriptive analysis includes (Mean, Standard deviation). The overall results and responses on this factor are illustrated below:

Table: 5.9

Descriptive result of Requirements Management

Item	Description	Mean	St. Deviation
1	C.1 Do you uniquely identify each requirement?	4.05	.686
2	C.2 Do you have defined policies for requirements management?	4.10	.788
3	C.3 Do you record requirements traceability from original sources?	4.10	.641

4	C.4 Do you define traceability policies?	3.75	.716
5	C.5 Do you maintain traceability manual?	3.90	.553
6	C.6 Do you use a database to manage requirements?	3.70	.865
7	C.7 Do you define change management policies?	3.55	.945
8	C.8 Do you identify global system requirements?	3.55	.759
9	C.9 Do you identify volatile requirements?	3.70	.733
10	C.10 Do you record rejected requirements?	3.80	.834
11	C.11 Do you reuse requirements over different projects?	4.00	.649

The values of mean and standard deviations of each item of Requirements Management are shown above. All participants conform that they practice Requirements Management either on regular bases or always. Items (1, 2, 3, and 11) scored the highest mean value (4.05, 4.10, 4.10, 4.00) respectively. Where others opinions are between 3.55 -3.90 and show a regular bases practice of Requirements Management as answered in items (4, 5, 6, 7, 8, 9, and 10). Where items 2 and 3 scored highest mean (4.10, 4.10)respectively. It means that IT staff in UUM defined policies for requirements management and record requirements traceability from original sources always.

The table(5.10) shows the frequencies of the item, the highest percentage of participants responses equal 55% on regular practice of this item, The result shows that the majority of participants practice this item regularly and it shows that IT staff in UMMIT have common agreement to practice this items on regular bases.

Table: 5.10
Do you uniquely identify each requirement

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	sometimes	4	20.0	20.0	20.0

regularly	11	55.0	55.0	75.0
always	5	25.0	25.0	100.0
Total	20	100.0	100.0	

See more in Appendix C

Requirements Management

The staffs in UUM IT department show high level of practices on requirements management and they fulfil these requirements always. These requirements found highly practiced by IT staffs are shown below:

- Uniquely identifying each requirement
- Defining policies for requirements management
- Recording requirements traceability from original sources
- Reusing requirements over different projects

This result is encouraging and shows that the staffs in UUM IT have high level of awareness on the importance of requirement management on IT project as part of overall RE practices.

Other practices associated with requirement management are found satisfactory and achieved by IT staffs on regular bases. These practices are shown below:

- Defining traceability policies
- Maintaining traceability manual
- Using a database to manage requirements
- Defining change management policies
- Identifying global system requirements
- Identifying volatile requirements

- Recording rejected requirements

5.4 Limitation

The study investigated three variables Requirements Description, Requirements Development, and Requirement Management. The scope of the study is limited to only three variables where other variables need for further investigation beyond the time limit of this research and availability of resources to investigate further variables such as requirement gathering, and analysis of software quality, and the effect of internal and external development in requirement engineering process. In addition to that the area of current study is limited to educational institutes, in particular IT department at UUM. The researcher suggests that other areas should be covered as well using the same variables, for example analysing RE practices in Small and Medium Enterprises, large corporation, government agencies and ministries.

5.5 Future Work

Although the findings of this study show the importance of Requirements Description, Requirements Development, and Requirement Management on RE practice in IT department and software development centres in educational institutes. Further investigation is required in the future to confirm and verify the results of this study by investigating Requirement Gathering, and Analysis of Software Quality in UUMIT. As a future work, the researcher aims to extend the scope of this study to other government and non-educational organizations and conducting further systematic literature analysis to cover more papers related to the topic of this research.

5.6 Summary of Chapter Five

The analysis of data in this chapter show that the current practice of requirement engineering in IT department of UUM is moderate and need for more improvement. Results show that IT staffs have a standards templates/documents for describing requirements and they have a specific lay out for the requirements document to improve readability, moreover the majority of participants do a regular practice of requirements elicitation, also the participants conform that they carry out a feasibility study before starting a new project always. The results also show that the majority of participants do a regular practice of requirements analysis and negotiation. It is found that IT staffs of UUM encourage the use of electronic systems to support requirements negotiations and define system boundaries always. All other opinions confirms high level of practice to requirements analysis and negotiation in IT department of UUM. The analysis shows that IT staffs always check that requirements document that meets their standards, and define validation checklists always and they defined policies for requirements management and record requirements traceability from original sources always.

CHAPTER SIX
CONCLUSIONS AND RECOMMENDATION

6.1 Introduction

The study analysed and evaluated the Requirements Engineering Practices among Software Developers at UUM Information Technology UUMIT. Requirement engineering composed of two main activities which are requirements development and requirements management. Bother these two phases are investigated in the IT department of UUM. The UUMIT is a centre that acts as a core of software development in UUM that deliver, support and maintain all the systems of the university. It is found that requirement phases are the bases for success of a software development life cycle.

Information Technology UUMIT has its own software developers that develop software in house to support the business function of UUM. This study aimed to investigate how the software developers at UUMIT practice the RE during software development. Therefore the focuses on how requirements are being elicited, analysed, negotiated and validated in requirements development of software projects organized by the staffs of UUMIT. Eliciting proper comprehension and management of requirements of software project are found as the main determinants of success in the process of development of software. In addition, from the requirements management activity this study focuses on how to handle changes in the requirements, version control of requirements, traceability and tracking of requirements. Therefore, this study did not consider other aspect or area in RE.

UUMIT has developed many software projects to support the university and its main business activities. The researcher investigated the current RE practices in this department during software development in order to reduce the development cost, delivering software on time as well as ensure the quality of software. For the present time, there are few studies to support RE practiced in UUMIT.

6.2 Conclusions

The results from the survey showed that the current practice of requirement engineering in IT department of UUM is encouraging but need for further development because most of RE practices associated with requirement development and requirement management are achieved on regular bases and not frequently. Therefore there is a need for more practices on these requirements by encouraging staffs to do them always and giving them training courses to increase their performance with RE practices in general.

The study also conducted systematic literature review (SLR) in order to make more accurate conclusion on RE practice in UUMIT. The primary studies are selected, the quality assessment used to include studies, the data extraction and monitoring is performed and the obtained data is synthesized. The conducting SLR possesses the following activities: selection of primary studies, data extraction, methodology, results and findings, and conclusions. The SLR used also to identify the sources for data and how researcher collected data in RE researches. The findings from SLR show that RE is defined by most of researchers as the process that develops product specifications that are complete, consistent and unambiguous. Scholars state that RE is considered the most imperative phase in software development process. The RE practices have been considered as a key issue that affects the success rate of projects in software industry.

6.3 Recommendations

Based on the results and findings of this study, the research sets the following recommendation:

- The staff in IT department need to get good requirements and effectively managing those requirements is a strong predictor of project success in software development.
- The needs for training staffs working in UUMIT on all requirements found practiced on regular bases to increase the rate of practices to higher level.
- Using any software development methodology that includes RE processes will lead to better results.
- The understanding of system requirements is critical to develop good systems for the university.

- The staff in IT department in UUM must give the same attention to all requirements of requirement development and requirement management and practice all requirements always to improve the RE process for outsourced software development projects.

- The staff in IT department in UUM can benefit from Agile to achieve software projects with high accuracy and in sorter time with less setbacks and errors during the implementation and development of software projects. Agile working software is the principal measure of progress in RE.
- RE in UUM need for Sustainable development to maintain a constant pace in software development.

REFERENCES

Abdullah, Nik Nailah Binti, Honiden, Shinichi, Sharp, Helen, Nuseibeh, Bashar, & Notkin, David. (2011). *Communication patterns of agile requirements engineering.* Paper presented at the Proceedings of the 1st workshop on agile requirements engineering.

Abran, Alain, Bourque, Pierre, Dupuis, Robert, & Moore, James W. (2001). *Guide to the software engineering body of knowledge-SWEBOK*: IEEE Press.

Alford, Mack W. (1977). A requirements engineering methodology for real-time processing requirements. *IEEE Trans. Software Eng., 3*(1), 60-69.

Anderson, Stuart, & Felici, Massimo. (2001). Requirements evolution from process to product oriented management *Product Focused Software Process Improvement* (pp. 27-41): Springer.

Aranda, Jorge, Easterbrook, Steve, & Wilson, Greg. (2007). *Requirements in the wild: How small companies do it.* Paper presented at the Requirements Engineering Conference, 2007. RE'07. 15th IEEE International.

Arayici, Yusuf, & Aouad, Ghassan. (2005). Computer integrated construction: an approach to requirements engineering. *Engineering, construction and architectural management, 12*(2), 194-215.

Arayici, Yusuf, Aouad, Ghassan, & Ahmed, Vian. (2005). Requirements engineering for innovative integrated ICT systems for the construction industry. *Construction innovation, 5*(3), 179-200.

Asghar, Sohail, & Umar, Mahrukh. (2010). Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components. *International Journal of Software Engineering, 1*(1), 32-50.

Basharat, Iqra, Fatima, Mamuna, Nisa, Rozina, Hashim, Rabia, & Khanum, Assia. (2013). *Requirements engineering practices in small and medium software companies: An empirical study.* Paper presented at the Science and Information Conference (SAI), 2013.

Bashir, M Salman, & Qureshi, M Rizwan Jameel. (2012). Hybrid Software Development Approach For Small To Medium Scale Projects: Rup, Xp & Scrum. *Cell, 966*, 536474921.

Bell, Douglas, Morrey, Ian, & Pugh, John. (1997). *The essence of program design*: Pearson Education India.

Bjarnason, Elizabeth, Wnuk, Krzysztof, & Regnell, Björn. (2011). *A case study on benefits and side-effects of agile practices in large-scale requirements engineering.* Paper presented at the Proceedings of the 1st Workshop on Agile Requirements Engineering.

Bland, J Martin, & Altman, Douglas G. (1996). Statistics notes: Measurement error. BMJ, 312(7047), 1654.

Boehm, W, B., Papaccio, & N, P. (1988). Understanding and Controlling Software Cost. *IEEE Transactions on Software Engineering*1462–1476.

Bokhari, Mohammad Ubaidullah, & Alam, Mahtab. (2013). Quality, Category and Characteristics of Secured Requirements. *International Journal of ICT and Management.*

Brooks, Frederik P, & Bullet, No Silver. (1987). Essence and accidents of software engineering. *IEEE computer*, *20*(4), 10-19.

Carr, Joseph J. (2000). Requirements engineering and management: the key to designing quality complex systems. *The TQM Magazine*, *12*(6), 400-407.

Cerpa, Narciso, & Verner, June M. (2009). Why did your project fail? *Communications of the ACM*, *52*(12), 130-134.

Chaos, Extreme. (2001). The Standish Group International: Inc.

Cheng, Betty HC, & Atlee, Joanne M. (2009). Current and future research directions in requirements engineering *Design Requirements Engineering: A Ten-Year Perspective* (pp. 11-43): Springer.

Cuevas, Gonzalo, Serrano, Al, & Serrano, Ar. (2004). *Assessment of the requirements management process using a two-stage questionnaire.* Paper presented at the Quality Software, 2004. QSIC 2004. Proceedings. Fourth International Conference on.

Damian, Daniela, Zowghi, Didar, Vaidyanathasamy, Lakshminarayanan, & Pal, Yogendra. (2004). An industrial case study of immediate benefits of requirements engineering process improvement at the Australian Center for Unisys Software. *Empirical Software Engineering, 9*(1-2), 45-75.

Daneva, Maya, & Ahituv, Niv. (2010). *A focus group study on inter-organizational ERP requirements engineering practices.* Paper presented at the Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement.

El Emam, Khaled, & Birk, Andreas. (2000). Validating the ISO/IEC 15504 measure of software requirements analysis process capability. *Software Engineering, IEEE Transactions on, 26*(6), 541-566.

El Emam, Khaled, & Madhavji, Nazim H. (1995). *A field study of requirements engineering practices in information systems development.* Paper presented at the Requirements Engineering, 1995., Proceedings of the Second IEEE International Symposium on.

Engineering, of, T. R. A., & Society. (2004). The Challenges of Complex IT Projects 1-45.

Felici, Massimo. (2005). Observational models of requirements evolution. University of Edinburgh.

Feller, William. (1950). An Introduction to Probability Theoty and Its Applications: Wiley.

Fernández, Daniel Méndez, & Penzenstadler, Birgit. (2014). Artefact-based requirements engineering: the AMDiRE approach. *Requirements Engineering*, 1-30.

Gause, Donald C, & Weinberg, Gerald M. (1989). *Exploring requirements: quality before* design: Dorset House New York.

George, D, & Mallery, P. (2003). Frequencies. SPSS for Windows step by step: A simple guide and reference, 11, 20-52.

Gliem, Joseph A, & Gliem, Rosemary R. (2003). *Midwest research to practice conference in adult, continuing and community education.* Paper presented at the Midwest Research-to-Practice Conference in Adult, Continuing and Community Education.

Grady, Robert B. (1999). *An economic release decision model: Insights into software project management.* Paper presented at the Proceedings of the Applications of Software measurement Conference.

Group, Standish. (1995). The CHAOS report. *línea], S. n., sl.*

Group, Standish. (2009). Chaos summary 2009. *Online report. Accessed June, 20.*

Hickey, Ann M, & Davis, Alan M. (2003). *Requirements elicitation and elicitation technique selection: model for two knowledge-intensive software development processes.* Paper presented at the System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on.

Hoare, CAR. (1981). The emperor's old clothes. Turing Prize address'. *Comm. ACM, 24*, 75-83.

Hoffmann, Matthias, Kuhn, Nikolaus, Weber, Matthias, & Bittner, Margot. (2004). *Requirements for requirements management tools.* Paper presented at the Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International.

Hull, Elizabeth, Jackson, Ken, & Dick, Jeremy. (2011). Requirements Engineering.

Insfran, Emilio, Chastek, Gary, Donohoe, Patrick, & do Prado Leite, Julio César Sampaio. (2013). Requirements engineering in software product line engineering.

Iqbal, Javed, Ahmad, Rodina, Nizam, Mohd Hairul, Nasir, Md, & Noor, Muhammad Asim. (2013). *Significant Requirements Engineering Practices for Software Development Outsourcing.* Paper presented at the Software Engineering Conference (ASWEC), 2013 22nd Australian.

Jain, Radhika, Cao, Lan, Mohan, Kannan, & Ramesh, Balasubramaniam. (2014). Situated Boundary Spanning: An Empirical Investigation of Requirements Engineering Practices in Product Family Development. *ACM Transactions on Management Information Systems (TMIS), 5*(3), 16.

Jebreen, Issam, & Wellington, Robert. (2013). *Understanding requirements engineering practices for packaged software implementation.* Paper presented at the Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on.

Kabaale, Edward, Kituyi, Geoffrey Mayoka, & Al-Mashari, Majed. (2015). A theoretical framework for requirements engineering and process improvement in small and medium software companies. *Business Process Management Journal, 21*(1).

Kassab, Mohamad. (2014). *An Empirical Study on the Requirements Engineering Practices for Agile Software Development.* Paper presented at the Software Engineering and Advanced Applications (SEAA), 2014 40th EUROMICRO Conference on.

Kavitha, CR, & Thomas, Sunitha Mary. (2011). Requirement gathering for small projects using agile methods. IJCA Special Issue on Computational Science-New Dimensions & Perspectives (3), 122-128.

Kenney, JF, & Keeping, ES. (1962). Moving averages. Kenney JF. Mathematics of statics. Princeton NJ: Van Nostrand.

Khan, Huma Hayat, Naz'ri bin Mahrin, Mohd, & bt Chuprat, Suriayati. (2014). Factors Generating Risks during Requirement Engineering Process in Global Software Development Environment. *International Journal of Digital Information and Wireless Communications (IJDIWC), 4*(1), 63-78.

Khankaew, Supha, & Riddle, Stephen. (2014). *A review of practice and problems in requirements engineering in small and medium software enterprises in Thailand.* Paper presented at the Empirical Requirements Engineering (EmpiRE), 2014 IEEE Fourth International Workshop on.

Kumar, S Arun, & Kumar, T Arun. (2011). Study the Impact of Requirements Management Characteristics in Global Software Development Project: An Ontology Based Approach. *International Journal of Software Engineering and Application, 2*(4).

Lauesen, S. (2002). Software Requirements: Styles and Techniques. *Harlow: Pearson Education Limited*.

Leffingwell, Dean. (1997). Calculating your return on investment from more effective requirements management. *American Programmer, 10*(4), 13-16.

Leffingwell, Dean, & Widrig, Don. (2000). *Managing software requirements: a unified approach*: Addison-Wesley Professional.

Leffingwell, Dean, & Widrig, Don. (2003). *Managing software requirements: a use case approach*: Pearson Education.

Lindholm, Ron. (2012). Defining requirements in a process and system development project.

Lindquist, Christopher. (2005). Fixing the requirements mess. *CIO Magazine*, 52-60.

Liskin, Olga, Schneider, Kurt, Fagerholm, Fabian, & Münch, Jürgen. (2014). *Understanding the role of requirements artifacts in kanban.* Paper presented at the Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering.

Liu, Julie Yu-Chih, Chen, Hun-Gee, Chen, Charlie C, & Sheu, Tsong Shin. (2011). Relationships among interpersonal conflict, requirements uncertainty, and software project performance. *International Journal of Project Management, 29*(5), 547-556.

Liu, Lin, Li, Tong, & Peng, Fei. (2010). *Why requirements engineering fails: A survey report from china.* Paper presented at the Requirements Engineering Conference (RE), 2010 18th IEEE International.

Mann, Prem S. (1995). Introductory Statistics. *Journal of the Royal Statistical Society-Series A Statistics in Society, 158*(2), 339.

Mendez Fernandez, D, Lochmann, Klaus, Penzenstadler, Birgit, & Wagner, Stefan. (2011). *A case study on the application of an artefact-based requirements engineering approach.* Paper presented at the Evaluation & Assessment in Software Engineering (EASE 2011), 15th Annual Conference on.

Mishra, Deepti, Mishra, Alok, & Yazici, Ali. (2008). *Successful requirement elicitation by combining requirement engineering techniques.* Paper presented at the Applications of Digital Information and Web Technologies, 2008. ICADIWT 2008. First International Conference on the.

Mohamed, Packeer, Farvin, Shafinah, Baharom, Fauziah, & Deraman, Aziz. (2013). AN EXPLORATORY STUDY ON CURRENT SOFTWARE DEVELOPMENT PRACTICES IN MALAYSIA FOCUSING ON AGILE BASED SOFTWARE DEVELOPMENT. *Science International, 25*(4).

Mohanani, Rahul, Ralph, Paul, & Shreeve, Ben. (2014). *Requirements fixation.* Paper presented at the Proceedings of the 36th International Conference on Software Engineering.

Niazi, Mahmood, & Babar, Muhammad Ali. (2007). De-motivators of software process improvement: An analysis of vietnamese practitioners' views *Product-Focused Software Process Improvement* (pp. 118-131): Springer.

Niazi, Mahmood, El-Attar, Mohamed, Usman, Muhammad, & Ikram, Naveed. (2012). *An empirical study identifying high perceived value requirements engineering practices in global software development projects.* Paper presented at the ICSEA 2012, The Seventh International Conference on Software Engineering Advances.

Nilofer, M, & Sheetal, G. (2012). Comparison of Various Elicitation Techniques and Requirement Prioritization Techniques. *International Journal of Engineering Research & Technology, 1*(3), 1-8.

Nuseibeh, Bashar, & Easterbrook, Steve. (2000). *Requirements engineering: a roadmap.* Paper presented at the Proceedings of the Conference on the Future of Software Engineering.

Pandey, Dhirendra, Suman, U, & Ramani, AK. (2010). Performance Measurement of Different Requirements Engineering Process Models: A Case Study. *International Journal of Computer Engineering & Technology (IJCET), 1*(2), 1-15.

Pandey, Dhirendra, Suman, Ugrasen, & Ramani, AK. (2011). *An Approach to Information Requirement Engineering.* Paper presented at the Information Science and Applications (ICISA), 2011 International Conference on.

Pfleeger, Shari Lawrence, & Atlee, Joanne M. (2006). *Software engineering: theory and practice*: Pearson Education India.

Pressman, Roger S. (2010). Software Engineering: A Practitioner's Approach, 7/e, Pressman & Associates: Inc.

Prior, Paul, & Keenan, Frank. (2005). *Requirements Management in a Distributed Agile Environment.* Paper presented at the WEC (2).

Quispe, Alcides, Marques, Maira, Silvestre, Luis, Ochoa, Sergio F, & Robbes, Romain. (2010). *Requirements engineering practices in very small software enterprises: A diagnostic study.* Paper presented at the Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the.

Rahman, Aedah Abd, Haron, Azlena, Sahibuddin, Shamsul, & Harun, Mazlan. (2014). An Empirical Study of the Software Project Requirements Engineering Practice in Malaysia Public Sector–A Perspective from the Stakeholders' Challenges. *International Journal of Computer Theory & Engineering, 6*(1).

Ralph, Paul, & Kelly, Paul. (2014). *The dimensions of software engineering success.* Paper presented at the Proceedings of the 36th International Conference on Software Engineering.

Ramingwong, Lachana. (2012). A REVIEW OF REQUIREMENTS ENGINEERING PROCESSES, PROBLEMS AND MODELS. *International Journal of Engineering Science & Technology, 4*(6).

Raygoza, Marc. (2003). *Visual SDLC: improving requirements engineering for object-oriented systems.* Paper presented at the Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications.

Rouibah, Kamel, & Al-Rafee, Sulaiman. (2009). Requirement engineering elicitation methods: A Kuwaiti empirical study about familiarity, usage and perceived value. *Information management & computer security, 17*(3), 192-217.

Swarnalatha k, Srinivasan, GN, Dravid, Meghana, kasera, Raunak, & Sharma, Kopal. (2014). A Survey on Software Requirement Engineering for Real Time Projects based on Customer Requirement. *International Journal of Advanced Research in Computer and Communication Engineering*.

Sadraei, Emila, Aurum, Aybüke, Beydoun, Ghassan, & Paech, Barbara. (2007). A field study of the requirements engineering practice in Australian software industry. *Requirements Engineering, 12*(3), 145-162.

Sankhwar, Shweta, Singh, Virendra, & Pandey, Dhirendra. (2014). REQUIREMENT ENGINEERING PARADIGM. *GLOBAL JOURNAL OF MULTIDISCIPLINARY STUDIES, 3*(3).

Savolainen, Juha, Kuusela, Juha, & Vilavaara, Asko. (2010). *Transition to agile development-rediscovery of important requirements engineering practices.* Paper presented at the Requirements Engineering Conference (RE), 2010 18th IEEE International.

Shah, Tejas, & V Patel, S. (2014). A Review of Requirement Engineering Issues and Challenges in Various Software Development Methods. *International Journal of Computer Applications, 99*(15), 36-45.

Sheikh, Javed Anjum, Dar, Hafsa Shareef, & Sheikh, Farzan Javed. (2014). Usability Guidelines for Designing Knowledge Base in Rural Areas *Design, User Experience, and Usability. User Experience Design for Everyday Life Applications and Services* (pp. 462-469): Springer.

Shubhamangala, BR, Rao, LM, Dakshinamurthy, A, & Singh, CGL. (2012). *Ability based domain specific training: a pragmatic solution to poor requirement engineering in CMM level 5 companies.* Paper presented at the Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on.

Sim, Susan Elliott, Alspaugh, Thomas A, & Al-Ani, Ban. (2008). *Marginal notes on amethodical requirements engineering: what experts learned from experience.* Paper presented at the International Requirements Engineering, 2008. RE'08. 16th IEEE.

Sommervile, & I. (2010). Software Engineering *Requirements Engineering* (pp. 1-25): Springer.

Sommerville, Ian, & Kotonya, Gerald. (1998). *Requirements engineering: processes and techniques*: John Wiley & Sons, Inc.

Sommerville, Ian, & Ransom, Jane. (2005). An empirical study of industrial requirements engineering process assessment and improvement. ACM Transactions on Software Engineering and Methodology (TOSEM), 14(1), 85-117.

Stangor, Charles. (2010). Research Methods for the Behavioral Sciences: Cengage Learning.

Suri, PK, Soni, Rachna, & Jolly, Ashish. (2009). Potential Effect of Creeping User Requirements on Project Management: A Simulation Approach. *IJCSNS, 9*(11), 256.

Tahir, Amjed, & Ahmad, Rodina. (2010). *Requirement engineering practices-an empirical study.* Paper presented at the International Conference on Computational Intelligence and Software Engineering (CiSE).

Talbot, Alison. (2011). *An investigation into requirements engineering current practice and capability in small and medium software development enterprises in New Zealand.* Auckland University of Technology.

Tarawneh, Feras, Baharom, Fauziah, Yahaya, Jamaiah Hj, & Zainol, Azida. (2011). *COTS software evaluation and selection: A pilot study based in Jordan firms.* Paper presented at the Electrical Engineering and Informatics (ICEEI), 2011 International Conference on.

Team, CMMI Product. (2002). Capability Maturity Model® Integration (CMMI SM), Version 1.1. *CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1. 1)*.

Team, CMMI Product. (2002). Capability Maturity Model® Integration (CMMI), Version 1.1--Continuous Representation.

Unterkalmsteiner, Michael, Feldt, Robert, & Gorschek, Tony. (2014). A taxonomy for requirements engineering and software test alignment. *ACM Transactions on Software Engineering and Methodology (TOSEM), 23*(2), 16.

Wahono, Romi Satria. (2003). *Analyzing requirements engineering problems.* Paper presented at the Proc of IECI Japan Workshop, ISSN.

Wiegers, Karl. (2013). *Creating a Software Engineering Culture*: Addison-Wesley.

Wiegers, Karl E. (2003). Software Requirements: Practical techniques for gathering and managing requirement through the product development cycle. *Microsoft Corporation*.

Winter, Robert, & Strauch, Bernhard. (2004). *Information requirements engineering for data warehouse systems.* Paper presented at the Proceedings of the 2004 ACM symposium on Applied computing.

Young, Ralph Rowland. (2004). *The requirements engineering handbook*: Artech House.

Zainol, Azida, & Mansoor, Sa'ad. (2008). *Investigation into requirements management practices in the Malaysian software industry*. Paper presented at the Computer science and software engineering, 2008 international conference on.

Zave, Pamela, & Jackson, Michael. (1997). Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology (TOSEM), 6*(1), 1-30.